

Universidade Federal do Pampa

Tiago Anunciação Ricaldi

Aplicação Móvel para a Consulta de Alvarás de Funcionamento

Alegrete

2016

Tiago Anuniação Ricaldi

Aplicação Móvel para a Consulta de Alvarás de Funcionamento

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Engenharia de Software.

Orientador: Prof^o. Dr. Cristiano Tolfo

Alegrete

2016

Ficha catalográfica elaborada automaticamente com os dados fornecidos
pelo(a) autor(a) através do Módulo de Biblioteca do
Sistema GURI (Gestão Unificada de Recursos Institucionais) .

R487a Ricaldi, Tiago Anuniação
Aplicação Móvel para a Consulta de Alvarás de Funcionamento
/ Tiago Anuniação Ricaldi.
112 p.

Trabalho de Conclusão de Curso(Graduação)-- Universidade
Federal do Pampa, ENGENHARIA DE SOFTWARE, 2016.
"Orientação: Cristiano Tolfo".

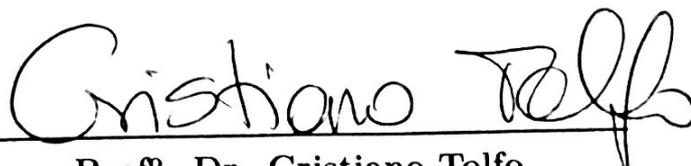
1. Alvarás de funcionamento. 2. Dispositivos móveis. 3.
Sistemas de informação. 4. Web Service. I. Título.

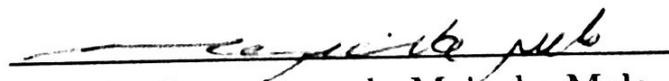
Aplicação Móvel para a Consulta de Alvarás de Funcionamento

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Engenharia de Software.

Trabalho de Conclusão de Curso defendido e aprovado em 20 de Junho de 2016

Banca examinadora:


Prof^o. Dr. Cristiano Tolfo
Orientador


Prof^a. Dra. Amanda Moíncke Melo
UNIPAMPA


Prof^o. M^c. Jean Felipe Patikowski
Cheiran
UNIPAMPA

Agradecimentos

Primeiramente agradeço aos meus pais Eloi Ricaldi e Eliane Ricaldi por todo amor e paciência prestados durante a vida e na realização dessa monografia.

A minha namorada Catieli Rigão que me aturou em diversos dias de stress. As minhas avós Laura e Jacira e a meu avô Heitor que sempre me apoiaram incondicionalmente.

Ao meu orientador Cristiano Tolfo, e aos professores Amanda Melo e Jean Cheiran que partilharam os seus conhecimentos comigo durante a trajetória deste trabalho e vida académica. Aos demais professores e ex professores do curso de Engenharia de software, que partilharam os seus conhecimentos sempre buscando o melhor aprendizado para os alunos.

Ao setor de divisão da tecnologia da informação da Prefeitura de Alegrete, que disponibilizou alguns dados para a contribuição do desenvolvimento desta monografia.

Aos colegas de curso e de faculdade, com aos quais tive o prazer de fazer amizades e conviver durante o curso.

Resumo

Os estabelecimentos públicos e privados têm utilizado sistemas de informação para dispositivos móveis a fim de fornecer informações úteis à população. O objetivo do presente trabalho é o desenvolvimento de um sistema para dispositivos móveis que disponibilize informações sobre a regularidade do alvará de funcionamento das organizações. Na metodologia para atender o objetivo inicialmente foi apresentada a proposta do sistema para potenciais usuários, empresários e representantes do órgão regulador, com os quais foram realizadas entrevistas informais e estórias com cartões de usuário com o intuito de identificar as funcionalidades do sistema. Na sequência foi desenvolvido um sistema para dispositivos móveis no sistema operacional Android integrado a um *web service*, onde os resultados alcançados foram o desenvolvimento de uma aplicação móvel que permita às pessoas consultar a situação dos alvarás de funcionamento dos estabelecimentos, e o desenvolvimento de um *web service* para consultar os dados dos alvarás que estão na base de dados de um órgão regulador. O estudo do estado atual do processo para a obtenção de informações de um alvará, para gerar o sistema proposto no processo do estado futuro até a construção da arquitetura indicaram a viabilidade da construção do sistema. A utilização dos cartões de estórias de usuários e entrevistas informais, foram importantes para perceber que o sistema é útil aos potenciais usuários. A validação do sistema com os potenciais usuários do sistema evidenciou que o sistema necessita de evoluções para melhorar a usabilidade do sistema. A contribuição desse trabalho, gera como legado estudos de iniciação científica que podem ser realizados para integrar essa aplicação a bancos de dados distribuídos e heterogêneos de diferentes órgãos reguladores; melhorar a usabilidade; realizar os testes em dispositivos reais para iniciar a operacionalização do sistema.

Palavras-chave: Alvarás de funcionamento. Dispositivos Móveis. Sistemas de informação.

Abstract

Public and private institutions have used information systems for mobile devices to provide useful information to the public. The aim of this work is the development of a system for mobile devices that provides information about the regularity of the business license of organizations. The methodology to meet the goal was initially presented the proposed system to potential users, businessmen and representatives of the regulatory body, with which were conducted informal interviews and stories with user cards in order to identify system features. Following a system was developed for mobile devices on the Android operating system integrated into a web service where the results were the development of a mobile application that allows people to check the status of the operating permits of establishments, and the development of a web service to query the data of licenses that are in the database of a regulator. The study of the current state of the process for obtaining information from a license to generate the system proposed in the process of the future state until the construction of the architecture indicated the viability of the system construction. The use of users and informal interviews stories cards were important to realize that the system is useful to potential users. The system validation with potential users of the system showed that the system needs changes to improve the system usability. The contribution of this work generates a legacy of scientific research studies that can be done to integrate this application to distributed databases and heterogeneous various regulatory bodies; improve usability; perform the tests on real devices to start the operation of the system.

Key-words: Permits. Mobiles. Information systems.

Lista de ilustrações

Figura 1 – Modelo de processo da metodologia.	23
Figura 2 – Arquitetura do Sistema Operacional Android.	28
Figura 3 – <i>Software Development Kit</i> (SDK) utilizados no Android Studio.	30
Figura 4 – Plugin Genymotion no Android Studio.	30
Figura 5 – Arquitetura cliente/servidor.	32
Figura 6 – Camadas do processo OpenUP	34
Figura 7 – Ciclo de vida do processo de <i>design</i> centrado no usuário	34
Figura 8 – Cartão de estória de usuário	35
Figura 9 – Modelo de processo das perspectivas de pesquisas dos trabalhos relaciona- onados.	37
Figura 10 – Fases do Processo Unificado Aberto (OpenUP).	43
Figura 11 – Processo do Estado Atual.	44
Figura 12 – Processo do Estado Futuro.	45
Figura 13 – Diagrama de casos de uso.	47
Figura 14 – Diagrama de Classes.	49
Figura 15 – Diagrama Relacional.	50
Figura 16 – Administração do Banco de Dados.	50
Figura 17 – Pontos de Melhorias sugeridos pelos usuários.	53
Figura 18 – Versões dos emuladores Android configurados para os testes.	54
Figura 19 – Cartões de estórias do usuário 1.	65
Figura 20 – Cartões de estórias do usuário 2.	66
Figura 21 – Cartão de estória do usuário 3.	66
Figura 22 – Cartões de estórias do usuário 4.	67
Figura 23 – Cartão de estória do usuário 5.	67
Figura 24 – Cartão de estória do usuário 6.	67
Figura 25 – Tela Principal.	73
Figura 26 – Pesquisa por Nome Fantasia da Organização.	74
Figura 27 – Tela Principal e Pesquisa por CNPJ, Nome ou Sigla.	75
Figura 28 – Pesquisa por GPS.	75
Figura 29 – Pesquisa por Categorias.	76
Figura 30 – Diagrama de sequência Pesquisar alvará por CNPJ, nome fantasia e sigla.	77
Figura 31 – Diagrama de sequência Pesquisar alvará por categoria.	78
Figura 32 – Diagrama de sequência Pesquisar alvará por GPS.	79
Figura 33 – Diagrama de sequência Mostrar dados do contato do alvará.	80
Figura 34 – Programas necessários para o desenvolvimento.	81
Figura 35 – Criação do Web Service AlvaraDAO passo 1.	83

Figura 36 – Criação do Web Service AlvaraDAO passo 2.	84
Figura 37 – Criação do Web Service AlvaraDAO passo 3.	85
Figura 38 – Criação do Web Service AlvaraDAO passo 4.	85
Figura 39 – Criação do Web Service AlvaraDAO Geração do XML passo 5.	86
Figura 40 – Criação do Web Service AlvaraDAO XML passo 6.	86
Figura 41 – Classe AlvaraDAO e visão geral do projeto.	87
Figura 42 – Método buscar alvará por Pesquisa	88
Figura 43 – Activity SearchResultActivity	89
Figura 44 – Telas de Funcionamento da funcionalidade Buscar por cnpj, nome fantasia e sigla	90
Figura 45 – Telas de Funcionamento de Alvará não encontrado	90
Figura 46 – Telas de Funcionamento Pesquisar por categoria órgão federal	91
Figura 47 – Método buscar alvará por Categoria	92
Figura 48 – Método buscar alvará por GPS	93
Figura 49 – Tela de Funcionamento Buscar Alvará por GPS	94
Figura 50 – Tela de Funcionamento Mostrar dados dos contatos do Alvará	95
Figura 51 – Tela de ajuda	96
Figura 52 – Adicionando o xml no software SoapUI.	97
Figura 53 – Inserção de dados.	97
Figura 54 – Atualização de dados.	98
Figura 55 – Exclusão de dados.	98
Figura 56 – Pesquisa Alvará por CNPJ.	99
Figura 57 – Pesquisa Alvará por Sigla.	99
Figura 58 – Pesquisa Alvará por Nome Fantasia.	100
Figura 59 – Pesquisa do alvará por categoria.	100
Figura 60 – Pesquisa do alvará por GPS.	101
Figura 61 – Pesquisa dos dados dos contatos alvará.	101
Figura 62 – Testes na versão 2.3.7 GingerBread	103
Figura 63 – Testes na versão Android 4.1.1 JellyBean	104
Figura 64 – Testes na versão Android 4.2.2 JellyBean	105
Figura 65 – Testes na versão Android 4.3 JellyBean	106
Figura 66 – Testes na versão Android 4.4.4 KitKat	107
Figura 67 – Testes na versão Android 5.0.0 Lollipop	108
Figura 68 – Testes na versão Android 5.1.0 Lollipop	109
Figura 69 – Testes na versão Android 6.0.0 Marshmallow	110

Lista de tabelas

Tabela 1 – Camadas do processo OpenUP.	33
Tabela 2 – Lista de requisitos funcionais e não funcionais do sistema.	46
Tabela 3 – UC01: Pesquisar alvará de uma organização pelo nome fantasia.	69
Tabela 4 – UC02: Pesquisar alvará de uma organização pelo CNPJ.	69
Tabela 5 – UC03: Pesquisar alvará de uma organização pela sigla.	70
Tabela 6 – UC04: Mostrar os dados do alvará da organização.	70
Tabela 7 – UC05: Pesquisar alvará de uma organização pela categoria de serviço.	70
Tabela 8 – UC06: Pesquisar o alvará de uma organização por GPS.	71
Tabela 9 – UC07: Mostrar os dados dos contatos dos alvarás.	71

Lista de siglas

API *Application Programming Interface*

AS-IS estado atual

GPS *Global Positioning System*

HTTP *Hyper Text Transfer Protocol*

IDE *Integrated Development Environment*

IHC Interface Humano-Computador

OpenUP Processo Unificado Aberto

SDK *Software Development Kit*

SO sistema operacional

SOAP *Simple Object Access Protocol*

SOs Sistemas operacionais

UDDI *Universal Description Discovery and Integration*

Web *World Wide Web*

WSDL *Web Services Descriptor Language*

XML *Extensible Markup Language*

Sumário

1	INTRODUÇÃO	21
1.1	Objetivos	22
1.2	Organização do Documento	22
2	METODOLOGIA	23
3	FUNDAMENTAÇÃO TEÓRICO-TECNOLÓGICA	25
3.1	Computação Móvel	25
3.2	Tecnologias da Computação Móvel	26
3.2.1	Características do Android	26
3.2.2	Características do Blackberry	26
3.2.3	Características do iOS	26
3.2.4	Características do MeeGo	27
3.2.5	Características do Windows Phone	27
3.2.6	Características do WebOS	27
3.2.7	Tecnologia Escolhida	27
3.3	Tecnologia Android	28
3.3.1	Arquitetura Android	28
3.3.2	Configuração do Ambiente Android	29
3.4	Web Service	30
3.4.1	Conceito de Web Service	30
3.4.2	Web Service e o Android	31
3.5	Processo de Desenvolvimento de Engenharia de Software	32
3.6	Processo de Interação Humano-Computador	33
3.7	Engenharia de Requisitos	35
3.7.1	Elicitação e Análise de Requisitos	35
3.7.2	Especificação de Requisitos	36
3.7.3	Validação de Requisitos	36
3.8	Considerações do Capítulo	36
4	TRABALHOS RELACIONADOS	37
4.1	Metodologia Utilizada na Revisão da Literatura	37
4.2	Trabalhos Relacionados	38
4.2.1	Primeira Perspectiva de Pesquisa	38
4.2.2	Segunda Perspectiva de Pesquisa	39
4.2.3	Terceira Perspectiva de Pesquisa	40

4.2.4	Análise dos Trabalhos Relacionados	40
4.3	Considerações do Capítulo	41
5	DESENVOLVIMENTO DO SISTEMA	43
5.1	Fase de Iniciação	43
5.1.1	O que construir?	44
5.1.2	Estado Atual e Estado Futuro	44
5.1.3	Elicitação e Análise	45
5.2	Fase de Elaboração	46
5.2.1	Especificação de Requisitos	47
5.2.2	Diagrama de Casos de Uso	47
5.2.3	Validação de Requisitos	47
5.2.4	Arquitetura	48
5.2.5	Diagrama de classes	48
5.2.6	Diagrama Relacional	50
5.2.7	Diagramas de Sequência	51
5.3	Fase de Construção	51
5.3.1	Primeira Iteração	51
5.3.2	Segunda Iteração	52
5.3.3	Terceira Iteração	52
5.4	Fase de Transição	52
5.4.1	Testes com Web Service	53
5.4.2	Validação com Potenciais Usuários	53
5.4.3	Teste de Compatibilidade	54
5.5	Considerações do Capítulo	55
6	CONSIDERAÇÕES FINAIS	57
	REFERÊNCIAS	59
	APÊNDICES	63
	APÊNDICE A – APÊNDICE A	65
	APÊNDICE B – APÊNDICE B	69
	APÊNDICE C – APÊNDICE C	73
	APÊNDICE D – APÊNDICE D	75
	APÊNDICE E – APÊNDICE E	77

APÊNDICE F – APÊNDICE F	81
APÊNDICE G – APÊNDICE G	83
APÊNDICE H – APÊNDICE H	87
APÊNDICE I – APÊNDICE I	97
APÊNDICE J – APÊNDICE J	103

1 Introdução

A computação móvel tem facilitado o acesso à informação às pessoas possibilitando a obtenção de informações públicas e privadas, melhorando consideravelmente o acesso à informação e promovendo uma mudança no modo de interação entre os seres humanos. Segundo [Gonçalves \(2012, p.16\)](#) “existe uma maior necessidade das pessoas em contar com o acesso a informações pessoais e corporativas, e isso fez com que a indústria trabalhasse mais para disponibilizar equipamentos que atendessem a essa necessidade”. Os dispositivos móveis como *smartphones* e *tablets* são formas de disseminar as informações às pessoas proporcionando diversos tipos de softwares que auxiliam nessa tarefa.

A motivação do presente trabalho é ajudar a população por meio de dispositivos móveis a obter informações sobre os alvarás de funcionamento dos estabelecimentos credenciados em um órgão regulador, e também auxiliar o órgão regulador na divulgação dessas informações para o público ter acesso aos estabelecimentos regularizados.

Atualmente as organizações públicas e privadas necessitam de um certificado de funcionamento para desempenhar as suas atividades de modo legalizado. Esse certificado é denominado alvará de funcionamento, sendo emitido pelo órgão regulador.

“O artigo 1º da lei nº 1.171, fica estabelecido que os estabelecimentos comerciais, industriais e institucionais somente poderão funcionar com alvará de funcionamento, expedido pela administração regional da circunscrição onde se localize. O primeiro parágrafo do artigo 1º da lei nº 1.171, comunica que o alvará de funcionamento é o documento hábil para que os estabelecimentos possam funcionar, respeitadas ainda as normas relativas a horário de funcionamento, zoneamento, edificação, higiene sanitária, segurança pública e segurança e higiene do trabalho e meio ambiente. O segundo parágrafo do artigo 1º da lei nº 1.171, menciona que exige-se um alvará de funcionamento para cada estabelecimento, inclusive para aqueles que gozem de imunidade ou isenção tributária no Distrito Federal, bem como para os que explorem atividades não lucrativas, mesmo que de caráter assistencial ou por prazo determinado. O terceiro parágrafo do artigo 1º da lei nº 1.171, relata que para o exercício de qualquer tipo de atividade econômica eventual, será exigido alvará de funcionamento com vigência correspondente ao período ou dias especificados”. ([BRASIL, 1996](#))

O trabalho realiza o desenvolvimento de um sistema para dispositivos móveis que disponibiliza informações públicas à população como consultar alvará dos bombeiros, consultar alvará de funcionamento, consultar alvará sanitário e ambiental, entre outros. O estudo de caso consiste no desenvolvimento de um sistema que forneça informações sobre os alvarás de funcionamento de organizações. O escopo do trabalho limita-se a realizar o desenvolvimento da aplicação consultando a base de dados de somente um órgão regulador, sendo assim o presente trabalho não aborda bases de dados heterogêneas.

1.1 Objetivos

O objetivo geral do trabalho é o desenvolvimento de um sistema para dispositivos móveis que disponibilize informações sobre a regularidade do alvará de funcionamento das organizações.

Os objetivos específicos são:

- A criação de um *web service* para consultar os dados cadastrados no banco de dados de um órgão regulador, e disponibilizar essas informações através do aplicativo.
- A criação de um aplicativo móvel que permita a consulta da regularidade do alvará de funcionamento das organizações.

1.2 Organização do Documento

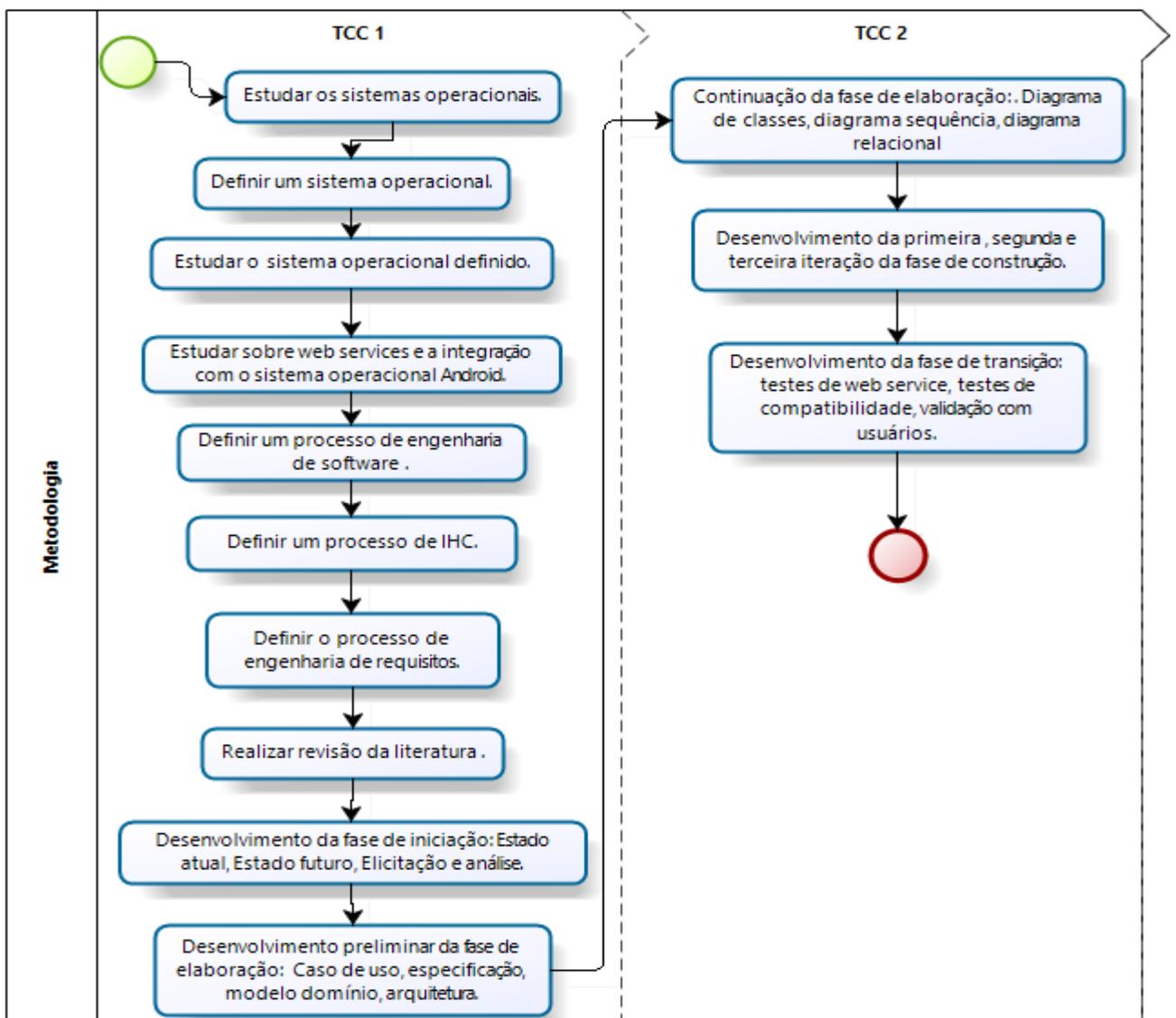
O texto está organizado como segue:

- No capítulo 2, da metodologia, é mostrado um modelo de processo com o conjunto de passos empregados na metodologia deste trabalho.
- No capítulo 3, da fundamentação teórico-tecnológica, são apresentadas as tecnologias da computação móvel e as práticas de engenharia de software empregadas no trabalho.
- No capítulo 4, dos trabalhos relacionados, é apresentada a metodologia empregada na revisão da literatura abordando as linhas de pesquisa que nortearam as pesquisas dos trabalhos relacionados.
- No capítulo 5, do desenvolvimento, são apresentados os resultados nas fases de iniciação, elaboração, construção e transição que foram desenvolvidas, baseado no processo de engenharia de software [OpenUP](#), integrado ao modelo de processo de Interface Humano-Computador (IHC) de *design* centrado no usuário.
- No capítulo 6, são apresentadas as considerações finais.

2 Metodologia

A proposta do sistema foi anunciada para os potenciais usuários, empresários e representantes do órgão regulador. E logo em seguida, foram aplicadas as técnicas de estórias com cartões de usuário e entrevistas informais, que foram utilizadas para descobrir os requisitos do sistema. A metodologia que foi aplicada para o desenvolvimento deste trabalho é mostrada no modelo de processo da [Figura 1](#).

Figura 1 – Modelo de processo da metodologia.



Com base na [Figura 1](#) é explicado o conjunto de passos aplicados na metodologia. Estudar os Sistemas operacionais (SOs) consiste em realizar um estudo para identificar

quais são os **SOs** da computação móvel que existem no mercado, pelo fato de que é necessário conhecer as características dessas plataformas para utilizar uma delas no desenvolvimento da aplicação deste trabalho; definir um sistema operacional (**SO**) é realizar a escolha de uma plataforma de desenvolvimento.

O **SO** escolhido é o Android porque ele é o mais acessível, pelo fato de ser uma plataforma completamente livre e de código aberto para realizar o desenvolvimento da aplicação; estudar o **SO** definido consiste em ver mais detalhadamente as características dele como a arquitetura do **SO** e a configuração do ambiente; estudar o *web service* tem o propósito de investigar os protocolos que realizam a troca de mensagens entre a aplicação proposta e a base de dados do órgão regulador; definir um processo de engenharia de software para o desenvolvimento do sistema é importante para a organização das tarefas.

O processo escolhido foi o **OpenUP**, porque ele pode ser empregado para equipes pequenas de desenvolvimento; definir um processo de **IHC** para aplicar em paralelo ao processo de engenharia de software é importante para desenvolver um aplicativo de acordo com as necessidades e expectativas do usuário. O modelo de processo de **IHC** empregado no desenvolvimento do trabalho é o de *design* centrado no usuário; definir o processo de engenharia de requisitos consiste em definir as técnicas que serão utilizadas nas fases de elicitação e análise, especificação e validação dos requisitos; revisar a literatura consiste em pesquisar os trabalhos relacionados com o presente trabalho.

Os trabalhos relacionados foram divididos em três perspectivas de pesquisa, na primeira perspectiva é realizada uma pesquisa de trabalhos relacionados a aplicativos móveis, que disponibilizam informações de utilidade pública; na segunda perspectiva é realizada uma pesquisa de aplicativos móveis que estão integrados a um *web service*; e na terceira perspectiva é realizada uma pesquisa dos trabalhos relacionados a aplicativos que fornecessem informações públicas, na loja da *Play Store* (**GOOGLE, 2015**).

O desenvolvimento das fases de iniciação, elaboração, construção e transição se referem às fases de desenvolvimento do sistema. Na fase de iniciação foram utilizadas as técnicas de cartões com histórias de usuários e entrevistas informais na elicitação dos requisitos. Nessa fase também é definida uma visão geral da solução e a organização das iterações e incrementos baseado no processo **OpenUP**. Na fase de elaboração, obtém-se uma visão mais consistente dos requisitos e também o desenvolvimento da arquitetura do sistema, diagrama de classes, diagrama de sequência, diagrama relacional. A fase de construção diz respeito à implementação das funcionalidades do sistema. Na fase de transição foram realizados os testes com *web service*, os testes de compatibilidade com diversas versões do **SO** Android, e também a validação do sistema com usuários.

3 Fundamentação Teórico-Tecnológica

Neste capítulo é apresentada a fundamentação teórica da aplicação desenvolvida neste trabalho, que tem por intuito identificar os principais conceitos, técnicas e processos de engenharia de software necessárias para o desenvolvimento do sistema. Esta seção está organizada da seguinte maneira: A seção 3.1 aborda sobre o crescimento da computação móvel. A seção 3.2 aborda de modo geral as características dos sistemas operacionais existentes na computação móvel para o desenvolvimento de aplicações. A seção 3.3 aborda o sistema operacional Android, a sua arquitetura e a configuração do ambiente. A seção 3.4 aborda sobre os conceitos de *web service* e a sua forma de comunicação com o sistema operacional Android. A seção 3.5 aborda o processo de desenvolvimento de engenharia de software *OpenUP*. A seção 3.6 aborda o processo de *IHC* de *design* centrado no usuário. A seção 3.7 aborda o processo de engenharia de requisitos empregado no presente trabalho. A seção 3.8 contém as considerações finais do capítulo.

3.1 Computação Móvel

A computação móvel obteve um crescimento exponencial nos últimos anos com a produção de diversas tecnologias inovadoras de *hardwares*, softwares e redes comunicações. Os autores, a seguir, relatam sobre o crescimento da computação móvel.

De acordo com [Lecheta \(2009, p.19\)](#) “existem mais de 3 bilhões de pessoas que possuem um aparelho celular. As pessoas procuram celulares com vários tipos de recursos como: câmeras, GPS, acesso à internet, etc”. A conectividade entre as pessoas através dos dispositivos móveis está aumentando cada vez mais em todos os cantos do mundo. Segundo [Costa e Filho \(2013, p.66\)](#) “atualmente os dispositivos móveis vêm se tornando cada vez mais populares, gerando o fenômeno conhecido como *mobile computing* (computação móvel)”. Ao longo dos últimos anos com a eclosão das tecnologias móveis acirrou a disputa entre os fabricantes de celulares, softwares e produtos eletrônicos de um modo geral. De acordo com [Junior et al. \(2014, p.1\)](#) “a concorrência no setor de teleequipamentos aumentou e alguns fabricantes, passaram a fabricar telefones celulares. A Apple propagou inovações no mercado de *smartphones* com o iPhone”.

O desenvolvimento de software para a computação móvel implica em conhecer e identificar as tecnologias existentes no mercado e escolher as que melhores se adaptam para o desenvolvimento de um produto. Neste contexto de desenvolvimento de aplicações móveis, diante da diversidade de linguagens de desenvolvimento, plataformas, bibliotecas, serviços de integração de tecnologias, *hardware*, é possível notar que o desenvolvimento de um sistema de informação para a computação móvel envolve diversas atividades a serem

realizadas.

3.2 Tecnologias da Computação Móvel

Existem diversos sistemas operacionais que podem ser empregados no desenvolvimento de sistemas de informação na computação móvel como: Android, Blackberry OS, iOS, MeeGo, Windows Phone, WebOS, etc.

3.2.1 Características do Android

O SO Android é o mais popular atualmente entre as pessoas que compram dispositivos móveis e pelos programadores que desenvolvem softwares para dispositivos móveis. Segundo [Lecheta \(2009, p.24\)](#) o “Android é uma plataforma, criada pela Google, completamente livre e de código aberto, e isso representa uma grande vantagem para a sua evolução, porque os programadores poderão contribuir na melhora da plataforma”. A contribuição da comunidade de desenvolvedores para a melhora da plataforma é relevante em termos de aprendizagem para o programador e também contribui para a evolução da plataforma. [Lecheta \(2009, p.23\)](#) relata que “a linguagem Java pode ser utilizada no desenvolvimento de aplicações no Android e a máquina virtual utilizada nesse sistema operacional é denominada Dalvik que é otimizada para dispositivos móveis”.

3.2.2 Características do Blackberry

[BlackBerry \(2015\)](#) “a empresa BlackBerry foi fundada em 1984 e sediada em Waterloo, possuindo escritórios na América do Norte, Europa, Ásia, Pacífico e América Latina”. As vantagens desse SO segundo [Costa e Filho \(2013, p.72\)](#) são “a execução de aplicativos em linguagem Java; envio e recebimento de dados em alta velocidade com tecnologia EDGE; e um teclado para comandar o sistema operacional”. As desvantagens desse SO para [Costa e Filho \(2013, p.72\)](#) consistem em “assistência técnica somente nas metrópoles; sistema proprietário; navegador de internet nativo”.

3.2.3 Características do iOS

Segundo [Apple \(2015\)](#), a empresa “faz o *hardware* e o sistema operacional do iPad, do iPhone e do iPod touch, e tudo é projetado para funcionar em conjunto”. Essa característica de produzir o *hardware* e o sistema operacional é bem peculiar da empresa. Segundo [Milani \(2012, p.16\)](#), “iOS é um SO diferente do padrão com características próprias, existindo algumas restrições como: Limite de processamento, tempo de resposta e o tamanho de memória”.

3.2.4 Características do MeeGo

O SO da Meego de acordo com a Intel (2015) surgiu através de uma fusão de duas plataformas na Intel e da Nokia. As vantagens dessa plataforma conforme a Intel (2015) são “código totalmente aberto, incentivo a participação da comunidade com as melhores práticas do modelo de desenvolvimento *open source*”. As desvantagens são para Costa e Filho (2013, p.73) “falta de um suporte de qualidade aos aplicativos; licenças proprietárias envolvendo Nokia e Intel na criação dos aplicativos”.

3.2.5 Características do Windows Phone

O Windows Phone foi lançado pela Microsoft, e conforme Monaco e Carmo (2012, p.6), “o objetivo era oferecer uma plataforma para os dispositivos, e não o *hardware* onde o sistema operacional é executado, aumentando a competitividade no mercado com aparelhos de diferentes fabricantes”. A vantagem desse SO para Costa e Filho (2013, p.70) “é que ele não está associado a um único *hardware*, existindo vários fabricantes podem licenciar aparelhos para acessar as funcionalidades e recursos do Windows Phone”. A desvantagem para esses autores Costa e Filho (2013, p.70) é que “a diversidade de fabricantes pode gerar um problema quanto aos pré-requisitos como *Wifi*, Tela *multitouch*, inviabilizando o funcionamento do Windows Phone”.

3.2.6 Características do WebOS

Segundo HP (2015), “a empresa abriu a plataforma webOS para a comunidade *open source* com o intuito de inovar essa plataforma centrada na *Web*”. As vantagens desse SO são para Costa e Filho (2013, p.74) “portabilidade para sistemas educacionais; touchpad e tela sensível; portabilidade para sistemas educacionais”. As desvantagens para Costa e Filho (2013, p.74) são “assistência técnica de alto custo; produto restrito ao mercado corporativo e participação pouco expressiva no mercado”.

3.2.7 Tecnologia Escolhida

O sistema operacional escolhido para o desenvolvimento da aplicação do presente trabalho é o Android. O motivo dessa escolha é porque esse sistema operacional, conforme (LECHETA, 2016), é o mais utilizado no desenvolvimento de softwares para *smartphones*, dominando o mercado de dispositivos móveis atualmente. O sistema operacional Android é apresentado mais detalhadamente na seção 3.3.

3.3 Tecnologia Android

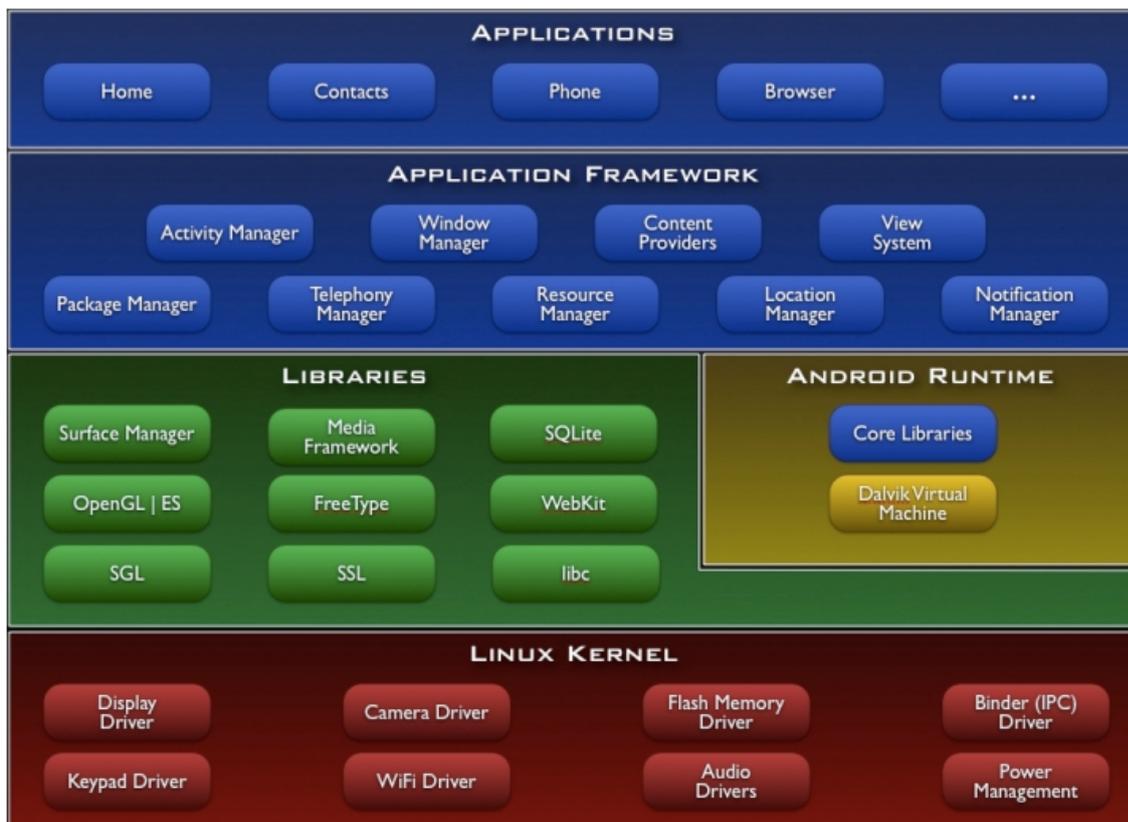
Segundo [Lecheta \(2009, p.21\)](#) o “Android é uma plataforma para aplicativos móveis como *smarthphones* e contém um sistema operacional baseado em Linux, interface visual rica, *Global Positioning System (GPS)*, e diversas aplicações já instaladas”.

3.3.1 Arquitetura Android

A arquitetura do sistema operacional Android é dividida quatro camadas que é possível visualizar nos itens abaixo e na [Figura 2](#).

- *Linux Kernel*
- *Libraries e Android Runtime*
- *Application Framework*
- *Applications*

Figura 2 – Arquitetura do Sistema Operacional Android.



Na camada *Linux Kernel* estão os *drivers* existentes como: *drivers* de áudio, câmera, *wifi*, entre outros. Segundo Santos (2013, p.16) na camada *Linux Kernel* “estão todos os *drivers* de baixo nível para os componentes de *hardware* do aparelho”.

Na camada das *Libraries* e *Android Runtime*, estão as bibliotecas que podem ser utilizadas pela aplicação. Segundo Santos (2013, p.16), essa camada “contém todo o código que disponibiliza as principais características do SO Android e essas bibliotecas podem ser consumidas pelas aplicações para ter as respectivas funcionalidades”. O *Android Runtime*, contém a máquina virtual Dalvik. Essa camada, para Santos (2013, p.16), “fornece um conjunto de instruções que permite que sejam escritas aplicações para Android em Java e também inclui a máquina virtual Dalvik que permite que cada aplicação rode seu processo”.

Na camada *Application Framework*, encontram-se os *frameworks* de aplicação que são utilizados pelos programadores durante o desenvolvimento de um aplicativo móvel. Segundo Santos (2013, p.17), essa camada “disponibiliza funcionalidades do SO para que possam ser utilizadas pelos desenvolvedores em suas aplicações”.

Na camada *Applications*, estão os aplicativos existentes no aparelho. Segundo Santos (2013, p.17), essa é a “camada superior que contém tanto as aplicações nativas do aparelho (contatos, navegador, etc.) como as instaladas posteriormente no aparelho”.

3.3.2 Configuração do Ambiente Android

O início de um desenvolvimento de uma aplicação requer uma configuração de ambiente antes de iniciar as atividades. O desenvolvimento do aplicativo é realizado através da *Integrated Development Environment (IDE)* Android Studio, que é a IDE oficial para o desenvolvimento de sistemas em android disponibilizado pela (GOOGLE, 2016). A escolha de um SDK é necessário para o desenvolvimento de sistemas em android, segundo Lecheta (2009, p.30) “o android SDK é o software utilizado para desenvolver aplicações no Android, que tem um emulador para simular o celular, ferramentas utilitárias e uma *Application Programming Interface (API)* completa à linguagem Java”. O SDK e a linguagem de desenvolvimento Java são utilizados no desenvolvimento do sistema deste trabalho. A Figura 3 mostra os SDK das versões android 6.0, 5.1.1, 4.2.2 e 4.0.3, que são utilizados no desenvolvimento do presente trabalho.

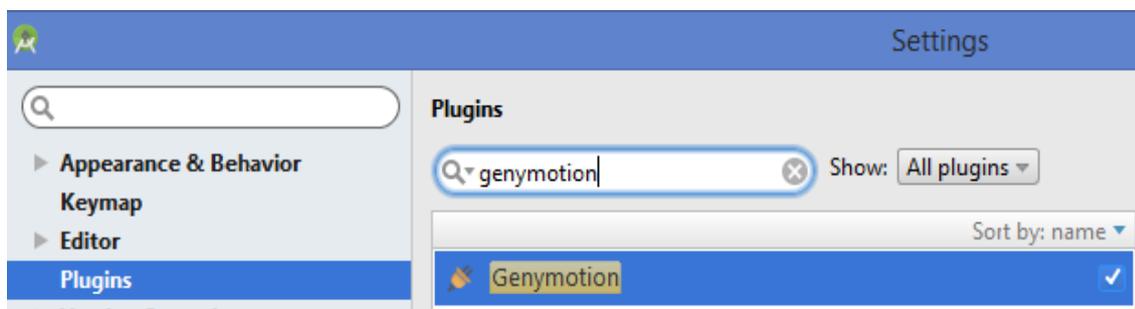
O emulador nativo do android studio não é utilizado neste trabalho, porque o seu desempenho de processamento ao inicializar o emulador do celular é limitado em relação a velocidade. Diante desse obstáculo, é adotado o emulador da (GENYMOTION, 2016) que possui uma gama de dispositivos com velocidade de processamento satisfatório, para testar as aplicações em android. A Genymotion (2016) pode ser integrada ao Android Studio através do seu plugin como mostra a Figura 4.

Figura 3 – SDK utilizados no Android Studio.

	Name	API Level	Revision	Status
<input checked="" type="checkbox"/>	Android 6.0	23	2	Installed
<input checked="" type="checkbox"/>	Android 5.1.1	22	2	Installed
<input checked="" type="checkbox"/>	Android 4.4.2	19	4	Installed
<input checked="" type="checkbox"/>	Android 4.0.3	15	5	Installed

Fonte: [Google \(2016\)](#)

Figura 4 – Plugin Genymotion no Android Studio.



Fonte: [Genymotion \(2016\)](#)

O plugin da Genymotion é configurado no Android Studio, e após a configuração é necessário escolher os emuladores para simular os aplicativos nas diferentes versões do Android como: GingerBread, JellyBean, KitKat, Lollipop e Marshmallow.

3.4 Web Service

Nessa seção será realizado um estudo específico sobre a tecnologia *web service*.

3.4.1 Conceito de Web Service

[Kieltyka \(2014, p.24\)](#) relata que um “*web service* é um tipo de aplicação para a *World Wide Web (Web)*, onde é possível executar uma rotina externa através de algum protocolo de rede”. [Freire \(2002, p.1\)](#) destaca que “a interoperabilidade é justamente um dos principais pontos de destaque dos web services”. [Lecheta \(2009, p.538\)](#) explica que o “*web service* é uma das tecnologias mais utilizadas para integrar aplicações”. Os *web services* promovem a troca de mensagens entre sistemas distintos promovendo uma comunicação entre eles. O *Hyper Text Transfer Protocol (HTTP)* e a *Extensible Markup Language (XML)* são utilizadas por *web services*. [Freire \(2002, p.1\)](#) cita que “os protocolos da arquitetura de um *web service* baseiam-se em padrões aceitos e adotados pelo mercado,

o protocolo [HTTP](#) como padrão para transporte e o [XML](#) como padrão de representação dos dados”.

As tecnologias que compõem a arquitetura dos *web services* para [DallOglio \(2014, p.28\)](#) são: “*Web Services Descriptor Language (WSDL)*, *Universal Description Discovery and Integration (UDDI)* e o *Simple Object Access Protocol (SOAP)*, unidas a [XML](#)”. [Freire \(2002, p.1\)](#) aborda que o [WSDL](#) “é a linguagem que descreve um *web service*, de forma que é possível para o cliente saber quais funcionalidades um dado serviço disponibiliza”. [DallOglio \(2014, p.28\)](#) cita que o [WSDL](#) “cria uma autodescrição pública do serviço e as informações que podem constar nesse documento são: O padrão da troca de mensagens, o protocolo de transporte e o endereço lógico do serviço”.

O [UDDI](#) para [Freire \(2002, p.1\)](#) “dita como *web services* podem ser registrado e localizados na rede”. [DallOglio \(2014, p.28\)](#) relata que o [UDDI](#) “É o diretório onde as informações dos *web services* podem ser armazenadas e esse diretório contém mecanismos que permitem a descoberta de serviços”.

O [SOAP](#) para [Freire \(2002, p.1\)](#) “é o coração da arquitetura de *web services*”. [DallOglio \(2014, p.28\)](#) explica que “é o protocolo adotado pela maioria dos *web services* para realizar a troca de mensagens. Baseado em [XML](#), o [SOAP](#) é projetado para que possa ser utilizado sobre qualquer protocolo”.

O [XML](#) para [Freire \(2002, p.1\)](#) “torna a representação dos dados independente de linguagem e de plataforma, e pode ser convertido para representações específicas para cada plataforma ou linguagem de programação”. [DallOglio \(2014, p.28\)](#) relata que o [XML](#) “é responsável pela interoperabilidade entre diferentes aplicações permitida pelos *web services* e serve de base para as outras três tecnologias dos *web services*: [SOAP](#), [UDDI](#) e [WSDL](#)”.

3.4.2 Web Service e o Android

A criação do serviço web service para acessar um banco de dados externo, no presente trabalho, necessita da instalação de alguns softwares como: A IDE ([ECLIPSE, 2016](#)) para realizar o desenvolvimento do web service na linguagem Java; o MySQL connector ([MYSQL, 2016](#)); o servidor Xampp para facilitar o acesso ao MySQL ([XAMPP, 2016](#)); o apache tomcat para poder hospedar o web service ([TOMCAT, 2016](#)); a biblioteca axis2 para trabalhar com web services, “ O apache Axis2 é um motor de Web Services/SOAP/WSDL, permitindo acessar serviços remotos” ([AXIS2, 2016](#)); o software SoapUI para realizar os testes com o web service ([SMARTBEAR, 2016](#)); e a biblioteca ksoap para poder consumir o web service a partir da aplicação Android ([KSOAP2, 2016](#)).

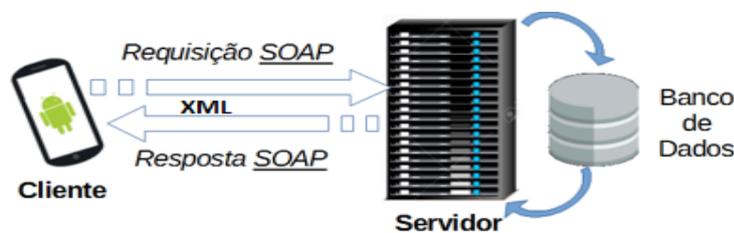
A integração de uma aplicação Android com um *web service*, depende de uma biblioteca que é responsável por consumir o *web service* a partir da aplicação do android.

Conforme Lecheta (2009, p.29) “o *framework* ksoap2 é uma biblioteca que consome *web services*, especialmente construída para dispositivos móveis com menos capacidade de processamento”. É necessário que se estabeleça um protocolo para realizar a comunicação entre a aplicação e o *web service*. Segundo Perumal et al. (2013, p.18) “o protocolo SOAP como tecnologia de serviços Web permite a troca de mensagens entre dois subsistemas diferentes, independentemente da plataforma operacional ou da linguagem utilizada”.

“Uma mensagem SOAP é um documento XML comum que contém: Um envelope que identifica o documento XML como uma mensagem SOAP; um cabeçalho que contém as informações de cabeçalho; um corpo que contém chamada e resposta de informações; um aviso contendo erros e informações de status”. (W3SCHOOLS, 2015)

A biblioteca ksoap é integrada ao projeto Android do presente trabalho e o protocolo SOAP será utilizado para realizar a comunicação entre a aplicação consumidora e o *web service*, onde será enviado uma mensagem SOAP como solicitação, e sendo retornada outra mensagem SOAP em um documento XML. Na Figura 5, é possível visualizar a arquitetura cliente/servidor, onde a aplicação Android é o lado do cliente e o *web service* é o lado do servidor que realiza a troca de mensagens entre a aplicação e a base de dados externa.

Figura 5 – Arquitetura cliente/servidor.



Fonte: Ricaldi e Tolfo (2015)

Essa arquitetura foi utilizada no desenvolvimento da aplicação e a troca de mensagens nesta arquitetura é realizada pelo protocolo SOAP que é representado por um documento XML que é uma linguagem de marcação extensível.

3.5 Processo de Desenvolvimento de Engenharia de Software

A escolha de um processo no desenvolvimento de software é importante para a organização e distribuição das tarefas ao longo de um projeto. Existem diversos processos como: Modelo cascata e alguns modelos de processos ágeis. Os modelos de processos ágeis incluem: Extreme Programming - XP (BECK, 2000); Scrum (SCHWABER, 2004); Processo Unificado Ágil (LARMAN, 2002); OpenUP (IBM, 2009), entre outros.

O **OpenUP** foi o escolhido para realizar a organização das tarefas do presente trabalho. Segundo a **IBM (2009)** “o OpenUP é um processo considerado Mínimo, Completo e Extensível, valorizando a colaboração entre a equipe e os benefícios aos interessados ao invés de formalidades desnecessárias”. As camadas do OpenUP estão explicadas na **Tabela 1**, baseadas em **Balduino (2007, p.7)** e **IBM (2009)**.

Tabela 1 – Camadas do processo OpenUP.

Camada Ciclo de vida do projeto	
Fases	Descrição dos objetivos das fases
Iniciação	Obter entendimento do que construir; identificação das funcionalidades chave do sistema; determinar ao menos uma solução possível e planejar e gerenciar iterações.
Elaboração	Obter um entendimento mais detalhado dos requisitos; projetar, validar uma arquitetura; planejar e gerenciar iterações.
Construção	Desenvolver iterativamente um produto completo que esteja pronto para transição para sua comunidade de usuários; planejar e gerenciar iterações.
Transição	Realizar testes e validações; planejar e gerenciar iterações.
Camada Ciclo de vida de iteração	
Descrição	
Consiste na divisão do projeto em iterações planejadas que podem variar de alguns dias a algumas semanas.	
Camada de micro incrementos	
Descrição	
Consiste na execução de um pequeno passo que deve ser mensurável para alcançar os objetivos de uma iteração, que pode representar o resultado de alguns dias ou horas de trabalho de uma pessoa.	

Na **Figura 6** é possível visualizar as camadas do processo OpenUP.

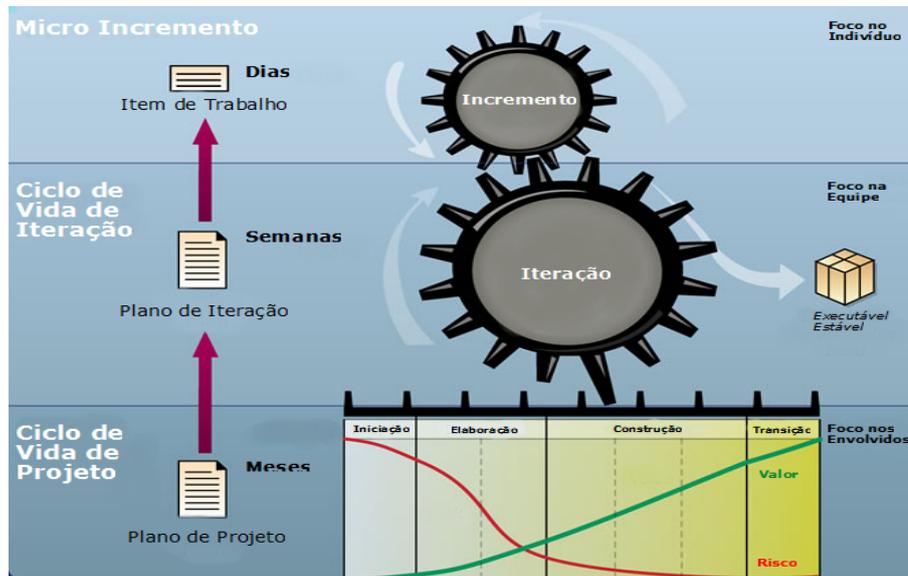
O processo de Engenharia de Software **OpenUP** é empregado no desenvolvimento do sistema, porque ele permite que o desenvolvimento dos incrementos das iterações seja realizado por equipes pequenas e até por um desenvolvedor. As necessidades dos usuários foram identificadas e os requisitos foram definidos com auxílio desse processo.

3.6 Processo de Interação Humano-Computador

Existem alguns modelos de processos de IHC (Interação Humano-Computador), conforme, (**BARBOSA; SILVA, 2010**), como: *Design* centrado no usuário, ciclo de vida em estrela, modelo de Schneiderman, engenharia de usabilidade.

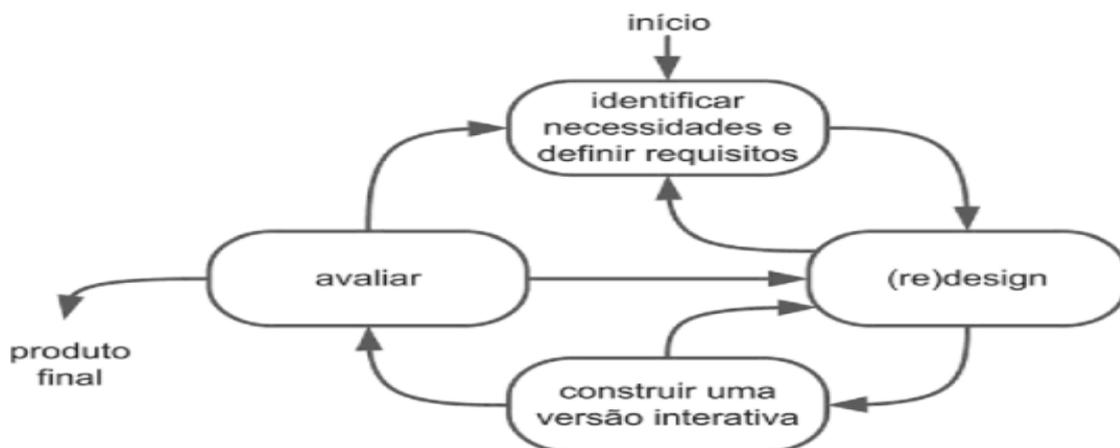
O modelo de processo de IHC de *design* centrado no usuário atua paralelamente ao processo de engenharia de software OpenUP, com a participação ativa dos potenciais usuários na construção dos protótipos do sistema. Segundo **Barbosa e Silva (2010, 101)** “Esse processo de design destaca a importância de avaliações da proposta de solução

Figura 6 – Camadas do processo OpenUP



Fonte: IBM (2009, Traduzido)

usando versões interativas e da iteração entre as atividades”. O ciclo de vida do processo de *design* centrado no usuário que atende através da participação as necessidades do usuário está representado na Figura 7.

Figura 7 – Ciclo de vida do processo de *design* centrado no usuário

Fonte: Barbosa e Silva (2010, p.102)

O processo de IHC de *design* centrado no usuário é empregado para realizar a validação dos requisitos do sistema com os potenciais usuários, através dos protótipos não funcionais. E também, na validação da versão interativa do sistema com os usuários, através dos protótipos funcionais.

3.7 Engenharia de Requisitos

O processo de engenharia de requisitos conforme [Sommerville \(2011, p.69\)](#), apresenta atividades como: “descobrir requisitos (elicitação e análise), convertendo-os em alguma forma padrão (especificação), verificar se os requisitos definem o que o cliente quer (validação)”.

3.7.1 Elicitação e Análise de Requisitos

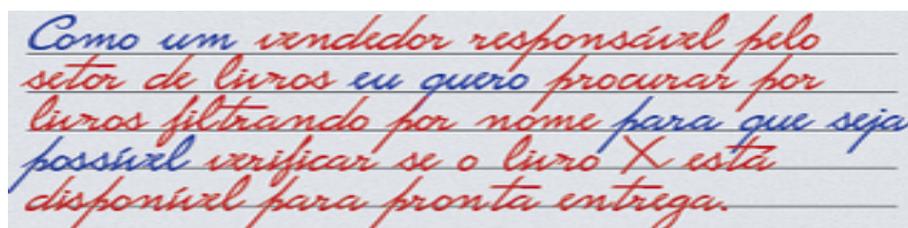
Na etapa de elicitação e análise conforme [Sommerville \(2011, p.69\)](#) “os engenheiros de software trabalham com usuários e clientes para obter informações sobre o domínio da aplicação, o serviços que o sistema deve oferecer”. Uma das técnicas que será utilizada no presente trabalho, e que [Sommerville \(2011\)](#) relata que existe para coletar os requisitos é a técnica de entrevistas informais com o usuário.

Segundo [Sommerville \(2011, p.44\)](#), “os requisitos são baseados em cenários ou em simples histórias de usuários, usadas como base para decidir a funcionalidade que deve ser incluída em um incremento do sistema”. [Cohn \(2004\)](#) relata que as histórias de usuários são criadas pelos usuários na etapa de levantamento de requisitos. O presente trabalho irá utilizar histórias de usuários para definir as funcionalidades do sistema.

Segundo [Pressman \(2011, p.88\)](#) escutar os usuários possibilita que o usuário crie diversas histórias “que descrevem o resultado, as características e a funcionalidade requisitada para o software a ser construído. Cada história é escrita em uma ficha pelo cliente”. Segundo [Cohn \(2004\)](#), uma história de usuário escrita no papel deve conter o formato: Como um (usuário)... Eu quero (ação)... Para(benefício esperado).

Na [Figura 8](#) é possível visualizar o exemplo de um cartão de história de usuário.

Figura 8 – Cartão de história de usuário



Fonte: [Rego \(2012\)](#)

No presente trabalho, o modelo de cartão de histórias de usuário e entrevistas informais serão empregados para realizar a descoberta dos requisitos do sistema.

3.7.2 Especificação de Requisitos

Na especificação de requisitos de acordo com [Sommerville \(2011, p.71\)](#) “os requisitos são documentados em documentos formais e informais de requisitos que podem ser produzidos”. No presente trabalho, será adotado o documento de especificação estruturada para descrever os requisitos.

3.7.3 Validação de Requisitos

Na validação de requisitos, [Sommerville \(2011, p.76-77\)](#) aborda que “é o processo pelo qual se verifica se os requisitos definem o sistema que o cliente realmente quer”. A validação dos requisitos na fase de análise, do presente trabalho, é realizada com a técnica de prototipação não funcional, conforme recomenda [Sommerville \(2011\)](#). Essa prototipação é realizada com o auxílio do modelo de processo de IHC de *design* centrado no usuário.

3.8 Considerações do Capítulo

O capítulo abordou sobre o crescimento da computação móvel nos últimos anos e apresentou alguns sistemas operacionais para o desenvolvimento de aplicações móveis, optando pelo sistema operacional Android para o desenvolvimento do presente trabalho. Estudou-se sobre *web service* e a sua forma de comunicação com o Android. O processo de engenharia de software *OpenUP* e o processo de Interação Humano-Computador de *design* centrado no usuário foram os processos estudados para realizar o desenvolvimento preliminar do presente trabalho. O processo de engenharia de requisitos foi abordado neste capítulo com a finalidade de aplicar os seus conceitos no desenvolvimento do trabalho.

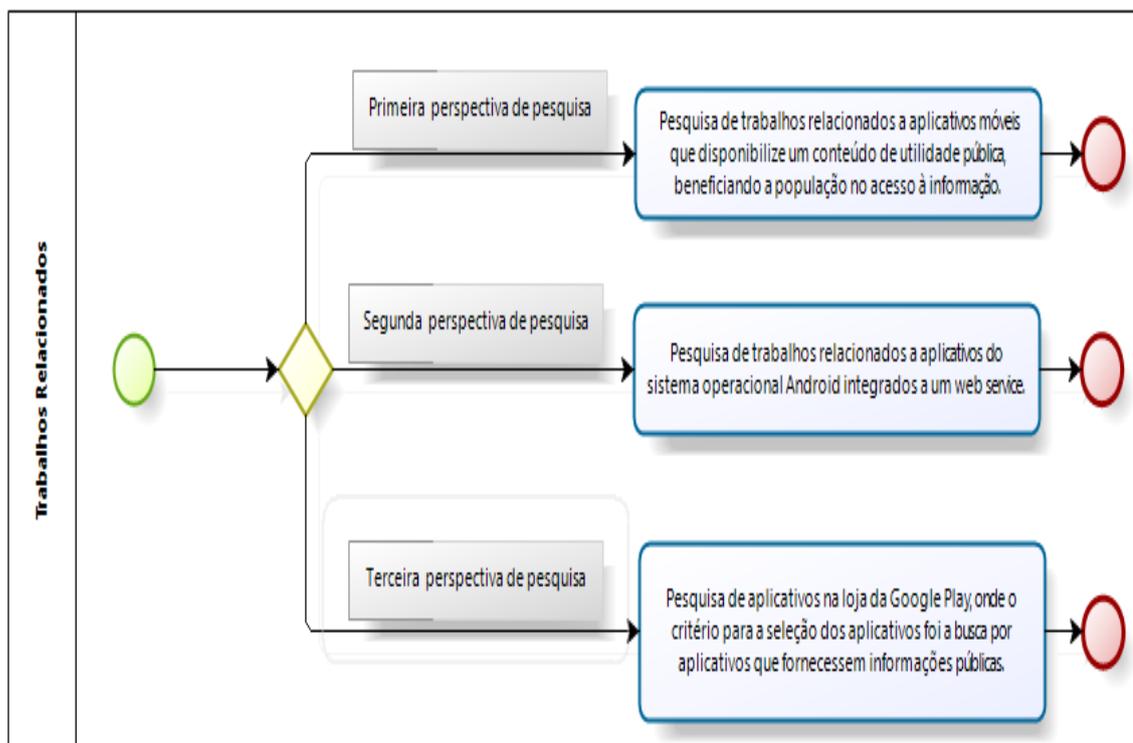
4 Trabalhos Relacionados

O presente capítulo aborda os trabalhos relacionados e a sua organização está da seguinte forma: A seção 4.1 aborda a metodologia utilizada nas pesquisas dos trabalhos relacionados. A seção 4.2 aborda os trabalhos relacionados. A seção 4.3 realiza uma consideração sobre o capítulo.

4.1 Metodologia Utilizada na Revisão da Literatura

Os trabalhos relacionados foram organizados em três perspectivas de pesquisa conforme o modelo de processo da [Figura 9](#) e nos itens abaixo:

Figura 9 – Modelo de processo das perspectivas de pesquisas dos trabalhos relacionados.



- A primeira perspectiva da pesquisa ocorreu através de um processo de revisão da literatura consultando as bases *Science Direct*, *ACM Digital Library*, *Google Acadêmico*. Nesse processo de revisão foram adotados alguns critérios para a seleção dos trabalhos, tais como: Data da publicação depois do ano de 2010; formato pdf; pesquisa de trabalhos relacionados a aplicativos móveis que disponibilizam um conteúdo de utilidade pública, beneficiando a população no acesso à informação. Na

pesquisa dos artigos as palavras-chave foram: Aplicações móveis para serviços públicos, desenvolvimento de aplicativos para serviços públicos, aplicações android para serviços públicos.

- A segunda perspectiva da pesquisa se deu através de um processo de revisão da literatura consultando as bases *Science Direct*, *ACM Digital Library*, Google Acadêmico. Nesse processo de revisão foram adotados alguns critérios para a seleção dos trabalhos: Data da publicação depois do ano de 2010; formato pdf; pesquisa de trabalhos referentes a aplicativos móveis do SO Android integrados a um *web service*. Na pesquisa dos artigos as palavras chaves foram: *web service*, Android com *web service*, desenvolvimento android integrado a *web service*, aplicativos com *web service*.
- A terceira perspectiva da pesquisa se deu através de uma procura de aplicativos na loja da *Play Store* (GOOGLE, 2015). O critério para a seleção dos aplicativos foi a busca por aplicativos que disponibilizassem informações públicas.

4.2 Trabalhos Relacionados

Nesta seção são apresentados os trabalhos relacionados referentes ao presente trabalho.

4.2.1 Primeira Perspectiva de Pesquisa

Na Pesquisa de trabalhos relacionados a aplicativos móveis que disponibilizam informações de utilidade pública foram encontrados os trabalhos dos autores: [Setiawan et al. \(2012\)](#), [Ferreira et al. \(2014\)](#), [Kim et al. \(2011\)](#) e [Cheng et al. \(2011\)](#).

[Setiawan et al. \(2012\)](#) propõem o desenvolvimento de um aplicativo móvel, que fornece informações sobre a localização de instalações públicas e rotas de transporte públicos. Nessa aplicação o objetivo é o usuário solicitar a localização de um lugar público e o aplicativo informa as melhores rotas com o auxílio de mapas para a pessoa achar o lugar desejado. A plataforma sugerida para o desenvolvimento da aplicação é o Android.

[Ferreira et al. \(2014\)](#) propõem um trabalho que tem por objetivo aumentar a qualidade dos serviços no setor de transporte público, na emissão de bilhetes por dispositivos móveis utilizando a plataforma Android. Esse aplicativo realiza serviços de pagamentos, validação de passagens, verificar preços de passagem. [Ferreira et al. \(2014, p.233\)](#) cita também outros serviços como: “acesso a informação em tempo real de informações de trânsito, horários e um planejador de viagem interativa”.

[Kim et al. \(2011\)](#) realizaram o desenvolvimento de um aplicativo móvel para *iPhone* que tem por objetivo facilitar a identificação de flores silvestres para estudantes e visitantes

de um parque natural. Esse aplicativo possui algumas funcionalidades que são segundo [Kim et al. \(2011, p.144\)](#) “mostrar mapa do parque, monitorar plantas raras, mostrar imagens das plantas, pesquisar as plantas por nome, mostrar o sistema de classificação das plantas”.

[Cheng et al. \(2011\)](#) realizaram o desenvolvimento de um aplicativo móvel utilizando a plataforma Android, que realiza o monitoramento da saúde das pessoas que têm doença pulmonar. Segundo [Cheng et al. \(2011, p.1\)](#) “o sistema permite que o dispositivo realize a vigilância da saúde da população com doenças pulmonares”. Essa vigilância acontece por meio de sensores que monitoram a transformação do movimentos realizados pelo corpo humano. Esse aplicativo garante o acesso à informação no que diz respeito a condição de saúde das pessoas, mas atinge uma parcela mais restrita da população, como os médicos e pacientes. Embora esse aplicativo atinja uma parcela mais restrita da população, a sua utilidade em termos de benefício às pessoas com doenças respiratórias é importante para o tratamento dos seus casos clínicos.

4.2.2 Segunda Perspectiva de Pesquisa

Na Pesquisa dos trabalhos relacionados a aplicativos móveis que estão integrados a um *web service*, foram encontrados os trabalhos dos seguintes autores: [Pocatilu \(2010\)](#), [Shanmugapriya e Tamilarasi \(2011\)](#), [DallOglio \(2014\)](#).

[Pocatilu \(2010, p.1\)](#) “apresenta os principais passos no desenvolvimento de uma aplicação *mobile learning* distribuído para o Android”. A Aplicação *mobile learning* de [Pocatilu \(2010\)](#) é uma aplicação móvel de aprendizagem que auxilia os alunos e os professores. Esse tipo de aplicação ajuda os alunos a submeter projetos, tarefas de casa, e também auxilia os professores a receber trabalhos, realizar a comunicação com os alunos, fazer o agendamento de atividades, etc. De acordo com [Pocatilu \(2010, p.1\)](#) “o aplicativo cliente se comunica com o servidor usando os serviços da *Web* que são consumidos usando uma biblioteca o kSOAP2, otimizado para Android”.

O trabalho de [Shanmugapriya e Tamilarasi \(2011\)](#) tem como objetivo o desenvolvimento de um aplicativo móvel no *SO* Android integrado a *web services* para desenvolver um ambiente interativo de aprendizagem que busque facilitar o aprendizado onipresente. Na integração do aplicativo móvel com o servidor a utilização da biblioteca kSOAP2 é recomendada para o desenvolvimento dessa aplicação. [DallOglio \(2014, p.14\)](#) tem como objetivo o desenvolvimento de uma aplicação utilizando o sistema operacional Android, que “permitirá aos usuários do ambiente de aprendizagem Univates virtual acessar o conteúdo e as atividades a partir de dispositivos móveis”. O desenvolvimento dessa aplicação utilizará segundo [DallOglio \(2014, p.15\)](#) “um cliente que utilize *web service* para acessar determinados conteúdos e atividade do univates virtual”. As tecnologias que compõem a estrutura de um *Web Service*, conforme [DallOglio \(2014, p.28\)](#) são: “WSDL, UDDI e

SOAP, unidas a acXML”.

O trabalho de [Ricaldi e Tolfo \(2015\)](#) é uma publicação relacionada ao desenvolvimento deste trabalho.

4.2.3 Terceira Perspectiva de Pesquisa

Na Pesquisa dos trabalhos relacionados a aplicativos que fornecessem informações públicas, na loja da *Play Store* ([GOOGLE, 2015](#)), foram encontrados os trabalhos dos seguintes autores: [Bombeiros \(2015\)](#), [Labdariu \(2015\)](#).

O aplicativo *Siat Mobile* do autor [Bombeiros \(2015\)](#) realiza a localização de eventos que estão autorizados pelo Corpo de Bombeiros. O aplicativo TRAFI, do autor [Labdariu \(2015\)](#), fornece dados sobre o trânsito de um local e também mostra informações sobre ônibus, metrô, lotação, entre outros. Esses aplicativos foram desenvolvidos para o sistema operacional Android.

4.2.4 Análise dos Trabalhos Relacionados

Os trabalhos fornecem diferentes tipos de informações às pessoas. Os trabalhos de [Pocatilu \(2010\)](#), [Shanmugapriya e Tamilarasi \(2011\)](#), [DallOglio \(2014\)](#) auxiliam alunos e professores em suas atividades diárias fornecendo um meio para que eles troquem informações referentes ao ensino como a entrega e recebimento de trabalhos, aulas a distância, etc. O trabalho de [Cheng et al. \(2011\)](#) é restrito a médicos e pacientes e realiza um monitoramento sobre o estado de saúde de pacientes com doenças pulmonares. Os trabalhos de [Setiawan et al. \(2012\)](#), [Ferreira et al. \(2014\)](#), [Kim et al. \(2011\)](#), [Bombeiros \(2015\)](#), [Labdariu \(2015\)](#) abrangem o público em geral.

Os trabalhos de [Setiawan et al. \(2012\)](#), [Pocatilu \(2010\)](#), [Shanmugapriya e Tamilarasi \(2011\)](#), [DallOglio \(2014\)](#) são trabalhos focados na aprendizagem sobre a comunicação do sistema operacional Android com um *web service*. Os trabalhos de [Bombeiros \(2015\)](#), [Labdariu \(2015\)](#) foram pesquisados com objetivo de mostrar que já existem aplicativos disponíveis na loja da ([GOOGLE, 2015](#)), e que essas aplicações auxiliam à população no seu cotidiano para obter acesso à informação.

Os trabalhos relacionados possuem uma ligação com a proposta deste trabalho porque ajudam a população a obter acesso à informação, via dispositivos móveis. E também pelo fato, de a maioria desses trabalhos utilizarem um *web service* integrado com o sistema operacional Android.

4.3 Considerações do Capítulo

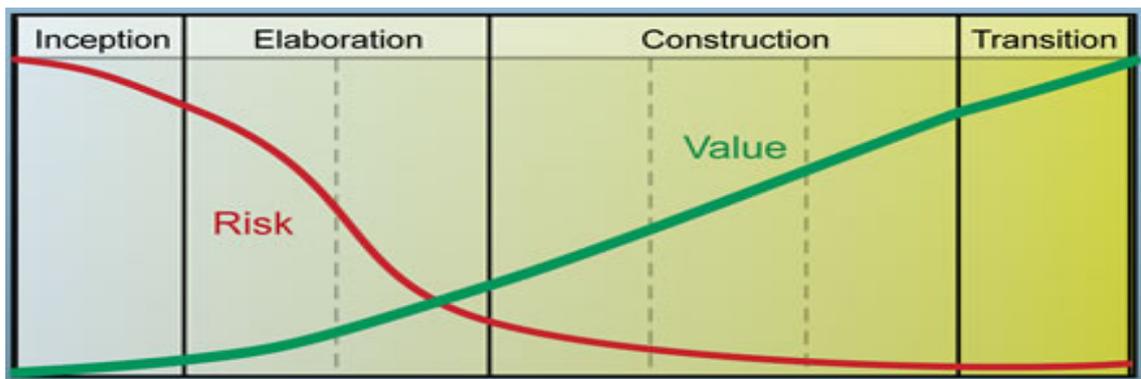
O capítulo abordou trabalhos que têm por objetivo proporcionar à população serviços que sejam úteis no seu cotidiano. Os trabalhos foram divididos em três perspectivas de pesquisa e durante o estudo deles foi definido que serão empregados para o desenvolvimento do sistema do presente trabalho o [SO](#) Android integrado a um *web service* que será consumido utilizando a biblioteca ksoap.

5 Desenvolvimento do Sistema

Os processos que auxiliam no desenvolvimento preliminar, conforme foram estudados na seção 3.5 e 3.6 do capítulo 3 são: **OpenUP** e *design* centrado no usuário. O capítulo está dividido na Seção 5.1 que aborda a fase de iniciação, na Seção 5.2 que aborda a fase de elaboração, na seção 5.3 que aborda a fase de Construção, na seção 5.4 que aborda a fase de transição e na Seção 5.5 que aborda as considerações do capítulo.

O processo **OpenUP** será instanciado através das suas fases, conforme **Figura 10**, a seguir.

Figura 10 – Fases do **OpenUP**.



Fonte: Eclipse (2009)

As fases da camada do ciclo de vida do projeto são desenvolvidas nas seções, a seguir.

5.1 Fase de Iniciação

No ciclo de vida de iniciação do processo **OpenUP** as atividades desenvolvidas de acordo com **Balduino (2007, p.7)** consistem em “entender o que vai ser construído; determinar ao menos uma solução possível; identificar funcionalidades chaves do sistema; planejar e gerenciar as iterações”. Na Seção 5.1.1 aborda o que construir. Na Seção 5.1.2 mostra uma visão geral da solução na **Figura 12** com o intuito de simplificar o processo atual da **Figura 11** na obtenção de informações sobre os alvarás. Na Seção 5.1.3 aborda as etapas de elicitação e análise de requisitos.

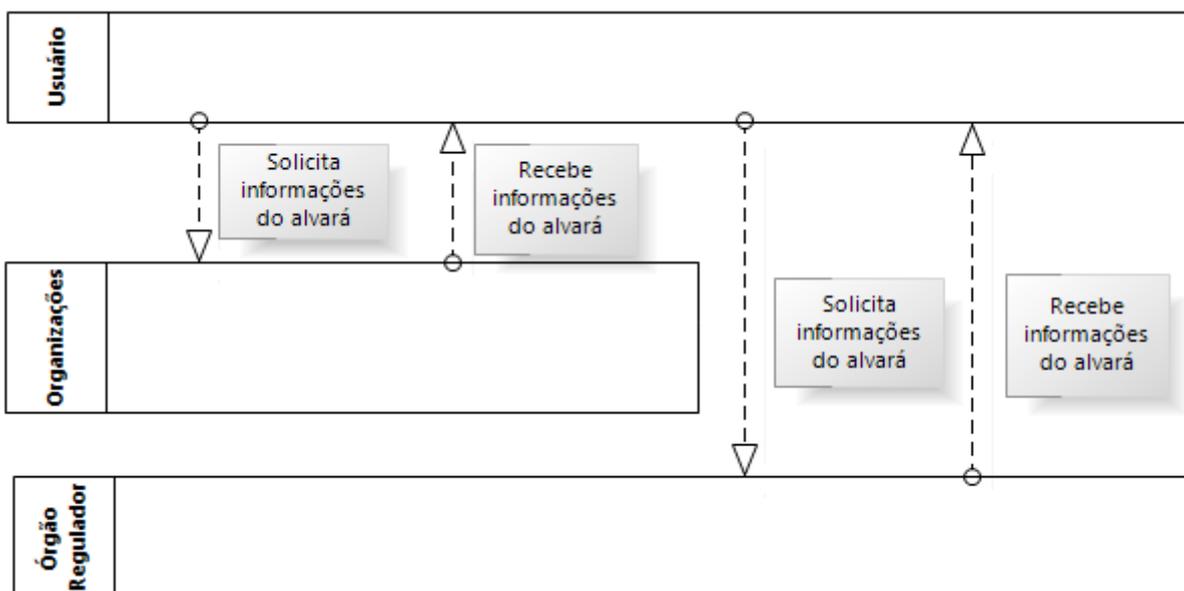
5.1.1 O que construir?

As pessoas que frequentam os estabelecimentos como restaurantes, casas noturnas, lojas comerciais, órgãos públicos, entre outros, muitas vezes possuem dificuldades para obter informações a respeito dos alvarás de funcionamento desses estabelecimentos. O alvará de funcionamento é um documento necessário para qualquer organização exercer as suas atividades de maneira regularizada, porque ele representa a liberação das organizações para trabalhar mediante a autorização do órgão regulador. Com base nessa dificuldade e na importância que o alvará representa para a legalização das atividades nas organizações, pretende-se realizar a construção de um sistema para dispositivos móveis, que auxilie na obtenção de informações a respeito da regularidade dos alvarás de funcionamento dos estabelecimentos.

5.1.2 Estado Atual e Estado Futuro

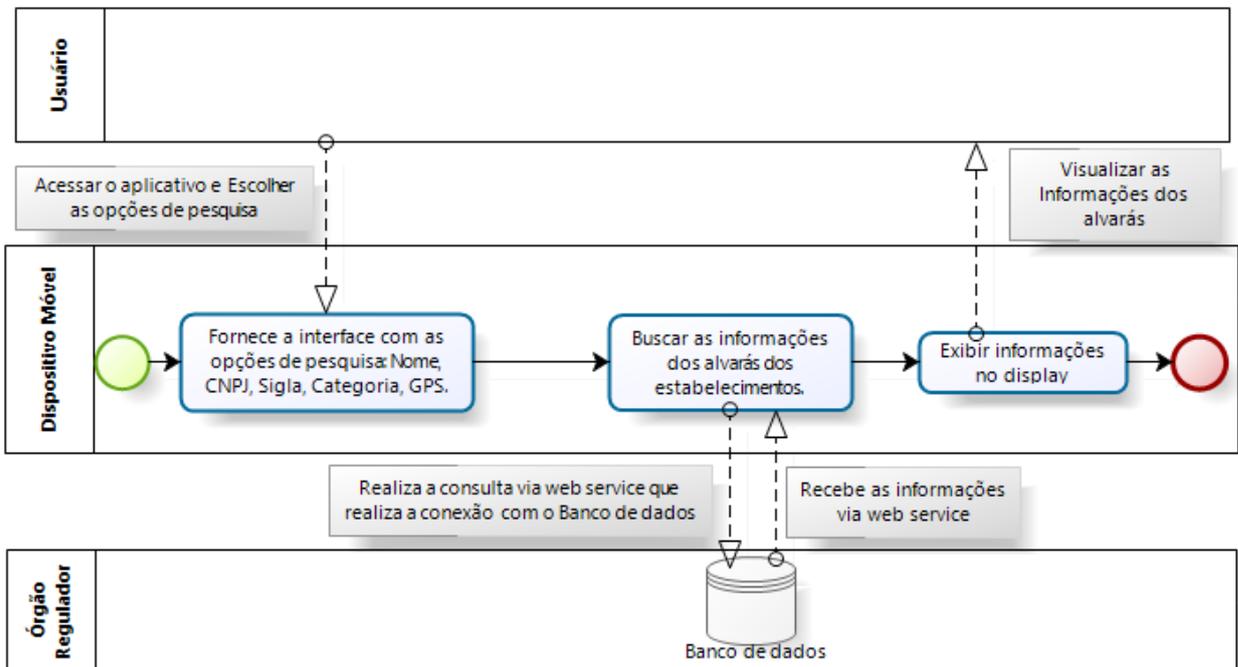
O processo do estado atual consiste em abstrair a maneira de como acontece a obtenção de informações a respeito dos alvarás de funcionamento atualmente, para que então seja possível propor uma solução que busque facilitar esse processo. Os atores identificados no estado atual (*AS-IS*) à obtenção de informações do alvará de funcionamento são: Usuário, empresa e órgão regulador. O processo do estado atual para a obtenção dessas informações é apresentado na [Figura 11](#) a seguir:

Figura 11 – Processo do Estado Atual.



O processo do estado futuro determina a construção de uma solução focada no desenvolvimento do sistema. A visão geral da solução construída está descrita no modelo de processo da [Figura 12](#) e identifica os seguintes atores: Usuário, sistema, órgão regulador.

Figura 12 – Processo do Estado Futuro.



A construção desse processo é importante, porque é possível obter uma visão geral da arquitetura utilizada no desenvolvimento do sistema.

5.1.3 Elicitação e Análise

Na elicitação de requisitos, conforme foi estudado na seção 3.7 de Engenharia de requisitos no capítulo 3, são empregados para realizar a descoberta dos requisitos do sistema as entrevistas informais recomendadas por Sommerville (2011) e o modelo de cartão de estórias de usuários de Cohn (2004).

A utilização do modelo de cartão de estórias de usuários foi utilizado com as partes interessadas no sistema como: Potenciais usuários, pessoas ligadas ao órgão regulador e empresários. As entrevistas informais foram aplicadas com um técnico administrativo da Unipampa e um empresário.

Os cartões de estórias de usuários foram aplicados em dois servidores da Unipampa, dois fiscais da prefeitura e dois empresários. As estórias de usuários coletadas estão no Apêndice A nas Figura 19, Figura 20, Figura 21, Figura 22, Figura 23, Figura 24.

Com base nas estórias de usuários, foi construída uma lista de requisitos funcionais e não funcionais do sistema. A Tabela 2 apresenta a lista de requisitos dividida em iterações e incrementos conforme recomenda o processo de desenvolvimento de Engenharia de software OpenUP.

Tabela 2 – Lista de requisitos funcionais e não funcionais do sistema.

Lista de Requisitos e Planejamento das Iterações			
Requisitos Funcionais	Requisitos Não Funcionais	Iteração	Incremento
RF 01: Pesquisar alvará de uma organização pelo nome fantasia com o auxílio do recurso de auto completar	Ao pesquisar o alvará de uma empresa, os dados gerais do alvará devem aparecer em menos de 3 segundos. (Requisito de Desempenho)	1ª	1
RF 02: Pesquisar alvará de uma organização pelo CNPJ com o auxílio do recurso de auto completar.	A interface deve ser minimalista para pesquisar o nome, CNPJ e Sigla (Requisito de usabilidade)	1ª	2
RF 03: Pesquisar alvará de uma organização pela sigla com o auxílio do recurso auto completar.	O mecanismo de busca deve considerar termos equivalentes. (Requisito de usabilidade)	1ª	3
RF 04: Mostrar dados do alvará da organização como: Razão Social, CNPJ, Nome fantasia, Ramo de atividade, endereço, horário de funcionamento, data de validade, número de cadastro do alvará, lotação máxima .	O sistema deverá acessar um banco de dados externo à aplicação Android para buscar os dados. (Requisito de interoperabilidade)	1ª	4
RF 05: Pesquisar alvará de uma organização por categoria de serviço.	A interface deve ser intuitiva e simular um celular Android para prover uma comunicação com o usuário. (Requisito de usabilidade)	2ª	1
RF 06: Pesquisar alvará de uma organização por GPS.	O sistema não pode ficar indisponível, sem conexão a internet. (Requisito de disponibilidade)	2ª	2
RF 07: Mostrar o contato dos alvarás como: Nome, Descrição, Telefone, Responsável, Email.	Ao pesquisar os contatos deve aparecer uma lista em menos de 3 segundos. (Requisito de Desempenho)	3ª	1

Os requisitos da lista estão divididos em três iterações. Na primeira iteração existem quatro incrementos, na segunda iteração existem dois incrementos e na terceira iteração existe um incremento a ser desenvolvido.

5.2 Fase de Elaboração

No ciclo de vida de elaboração do processo [OpenUP](#) as atividades a serem desenvolvidas segundo [Balduino \(2007, p.7\)](#) consistem em “obter um entendimento mais detalhado dos requisitos; projetar, validar uma arquitetura; planejar e gerenciar de iterações”. A seção está organizada do seguinte modo: Na seção 5.2.1 especificação de requisitos, que consiste na documentação detalhada e estruturada dos requisitos gerados a partir das histórias de usuários. Na seção 5.2.2 define o diagrama de casos de uso. Na seção 5.2.3 validação de requisitos, que consiste na criação dos protótipos não funcionais para verificar com o usuário se os requisitos gerados estão de acordo com a sua vontade. Na seção 5.2.4 define a arquitetura que será utilizada na implementação do sistema. Na seção 5.2.5 define o diagrama de classes da aplicação. Na seção 5.2.6 define o diagrama relacional da

aplicação. Na seção 5.2.7 define o diagramas de sequência das funcionalidades.

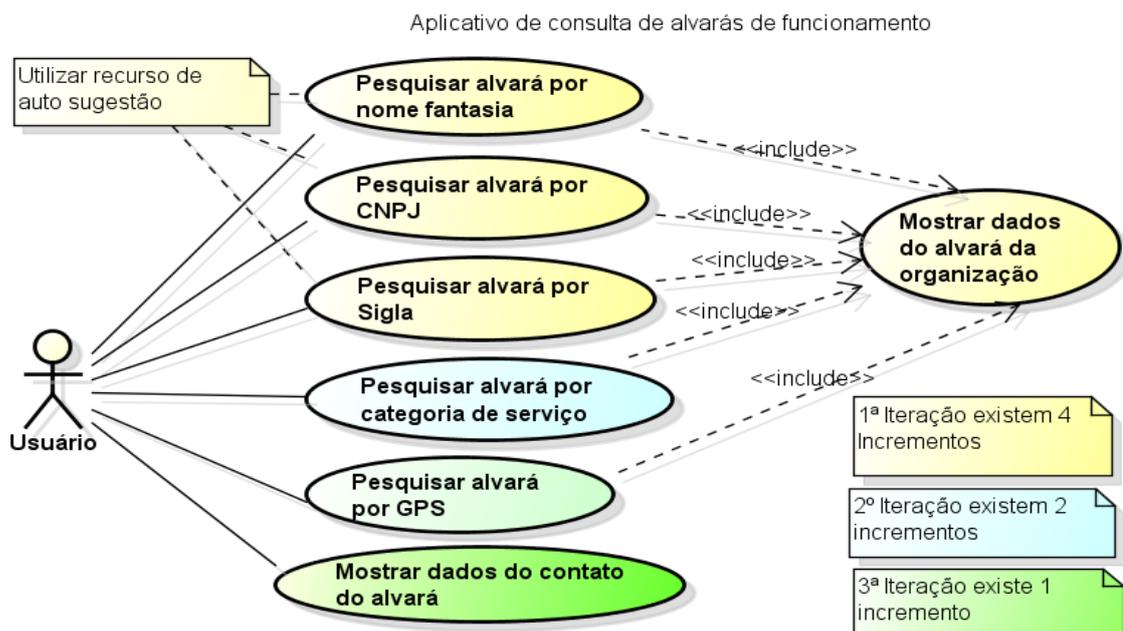
5.2.1 Especificação de Requisitos

A especificação de requisitos permite obter um entendimento mais detalhado dos requisitos, conforme foi estudado na seção 3.7 de engenharia de requisitos no capítulo 3, é a documentação formal ou informal dos requisitos gerados. A documentação estruturada da descrição dos requisitos estão no Apêndice B nas [Tabela 3](#), [Tabela 4](#), [Tabela 5](#), [Tabela 6](#), [Tabela 7](#), [Tabela 8](#), [Tabela 9](#).

5.2.2 Diagrama de Casos de Uso

Os casos de uso que estão representados na [Figura 13](#) mostram as interações que o usuário irá realizar com o sistema, e também as iterações com os seus respectivos incrementos.

Figura 13 – Diagrama de casos de uso.



O diagrama de casos de uso foi gerado a partir da especificação de requisitos e serve para poder compreender e visualizar as funcionalidades que serão disponibilizadas através do aplicativo.

5.2.3 Validação de Requisitos

A validação de requisitos, conforme foi estudado na seção 3.7 de engenharia de requisitos no capítulo 3, de acordo com [Sommerville \(2011, p.76-77\)](#) “é o processo pelo

qual se verifica se os requisitos definem o sistema que o cliente realmente quer. As técnicas de validação de requisitos são: Revisão de requisitos, prototipação, geração de casos de testes”. Com base nessa definição para realizar a validação dos requisitos da fase de análise foram desenvolvidos protótipos não funcionais, baseado no modelo de processo de IHC de *design* centrado no usuário.

A primeira versão dos protótipos não funcionais produzidos estão no Apêndice C nas [Figura 25](#), [Figura 26](#). A primeira versão desses protótipos foram validados com alguns potenciais usuários, que solicitaram algumas alterações nas interfaces. As partes interessadas no sistema fizeram sugestões para deixar a interface mais minimalista.

A partir das sugestões foi construído a segunda versão dos protótipos não funcionais representados no Apêndice D nas [Figura 27](#), [Figura 28](#), [Figura 29](#). A segunda versão dos protótipos também foram validados com os potenciais usuários que fizeram algumas observações à respeito dos botões quadrados com sugestões para arredondá-los, e também no que concerne a falta de informações na funcionalidade do GPS sobre algum alvará que o usuário esteja procurando e não o encontra, ocorreram algumas sugestões para colocar uma mensagem para o usuário como: “Seu local não foi encontrado ?” “ Entre em contato com o órgão regulador ou ligue para órgão regulador ou envie uma mensagem ao órgão regulador”.

5.2.4 Arquitetura

A aplicação do presente trabalho possui um banco de dados MySQL, que consiste na base de dados de um órgão regulador, e para fazer essa comunicação do dispositivo móvel com a base de dados, será utilizado um *web service* que será hospedado por um servidor. Essa base de dados é externa ao aplicativo Android, e refere-se a base de dados do órgão regulador que contém todas as informações necessárias e atualizadas sobre os dados dos alvarás dos estabelecimentos. A arquitetura da aplicação está representada na [Figura 5](#) na seção 3.4.2 do capítulo 3 da Fundamentação Teórica.

5.2.5 Diagrama de classes

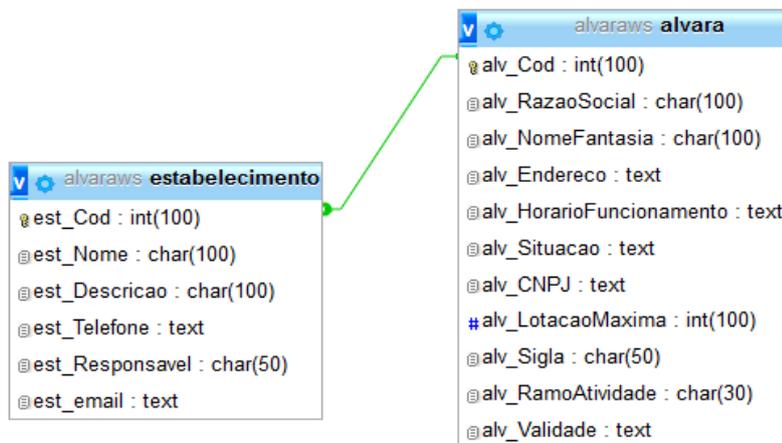
O diagrama de classes representado na [Figura 14](#) é o refinamento da arquitetura. O modelo foi dividido nos pacotes: *Activity* (camada view), *Model* (camada de negócio), *Serviço Web Service-tcc* (Camada de serviço *Web*) e *Dao* (camada do aplicativo que realiza a conexão com web service), *Provider* (pacote das recentes sugestões).

A camada *Activity*s possui uma dependência da camada de negócio e vice-versa. A camada de negócio possui uma dependência com a Camada de *Dao* do aplicativo e vice-versa. A camada *Dao* do aplicativo possui uma dependência com a camada *tcc* do serviço *web service*, onde o protocolo SOAP realiza a comunicação entre elas. O serviço *web*

5.2.6 Diagrama Relacional

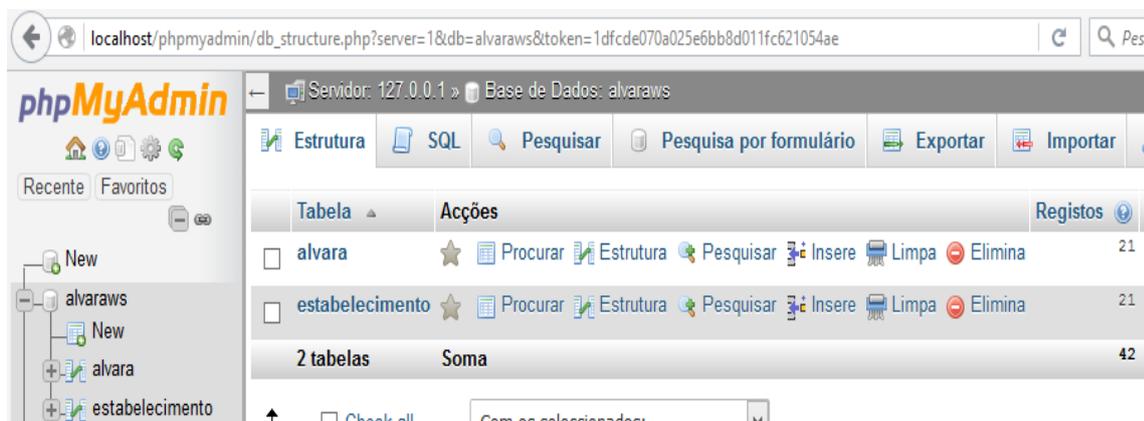
O diagrama relacional do banco de dados MySQL, denominado alvaraws, representado na Figura 15, possui as entidades alvara e estabelecimento. Na tabela alvara estão contidos os atributos de um alvará e na tabela estabelecimento estão contidos os atributos do contato do alvará.

Figura 15 – Diagrama Relacional.



A administração do banco de dados é realizada no phpMyAdmin, conforme a Figura 16. O phpMyAdmin facilita para a manipulação do banco de dados, onde é possível inserir as entidades e as suas operações de inserir, atualizar, deletar e pesquisar dados.

Figura 16 – Administração do Banco de Dados.



Nas entidades alvará e estabelecimento ao atualizar dados e deletar dados ocorre um efeito cascata, porque ao digitar o idAlvara para realizar as operações de atualizar ou deletar algum alvará, as tabelas sofrem atualização ou exclusão juntas em modo cascata.

5.2.7 Diagramas de Sequência

Os diagramas de sequência representando a interação entre os objetos, que trocam mensagens entre si, estão no Apêndice E, na [Figura 30](#), [Figura 31](#), [Figura 32](#), [Figura 33](#).

5.3 Fase de Construção

No ciclo de vida de construção do processo [OpenUP](#) as atividades a serem desenvolvidas segundo [Balduino \(2007, p.7\)](#) consiste em “realizar a implementação das iterações do sistema e de um produto completo”. A seção está organizada do seguinte modo: Na seção 5.3.1 ocorre o desenvolvimento da primeira iteração. Na seção 5.3.2 acontece o desenvolvimento da segunda iteração e na seção 5.3.3 ocorre o desenvolvimento da terceira iteração.

5.3.1 Primeira Iteração

O início desta iteração consistiu em realizar o *download* de todos os programas e bibliotecas que são necessárias para o início do desenvolvimento, que foram citadas na seção 3.4.2 Web Service e o Android, no capítulo 3, da fundamentação teórica. Na [Figura 34](#), no Apêndice F é possível visualizar os softwares necessários para realizar o desenvolvimento.

O procedimento inicial do desenvolvimento consistiu em realizar a criação do *web service* para acessar uma base de dados externa. A criação do *web service* pode ser visualizada na sequência das [Figura 35](#), [Figura 36](#), [Figura 37](#), [Figura 38](#), [Figura 39](#), [Figura 40](#) no Apêndice G. A [Figura 40](#) é o documento XML que foi gerado para a representação dos dados, nesse documento está contido todos os métodos das funcionalidades necessárias para realizar o acesso ao banco de dados externo MySQL para inserir, atualizar, consultar e deletar os dados de um alvará. Os testes com o *web service* e o banco de dados foram realizados através do software SoapUI na seção 5.4.1 na fase de transição.

O desenvolvimento da 1ª iteração do aplicativo na plataforma Android, iniciou-se depois da criação do *web service*. Essa iteração consiste em desenvolver a pesquisa de um alvará por cnpj, nome fantasia, sigla com o recurso de auto sugestão e mostrar os dados do alvará na Aplicação Android. O procedimento realizado para integrar o aplicativo ao servidor web service, conforme a seção 3.4.2, é a adição da biblioteca ksoap2 ao projeto. A [Figura 41](#) mostra uma visão geral dos pacotes do projeto e da classe AlvaraDAO, no apêndice H.

O método que concerne a essa iteração é o que busca um alvará por pesquisa de cnpj, nome fantasia e sigla. Esse método está explicado na [Figura 42](#), no Apêndice H. O requisitos não funcionais de auto sugestão e de pesquisar por termos equivalentes foram

implementados e são explicados na [Figura 43](#). As telas do funcionamento do sistema nesta iteração e na outras utilizam, como exemplo, o nome da Universidade Federal do Pampa, campus Alegrete, e podem ser visualizadas na [Figura 44](#). Na [Figura 45](#) mostra quando um usuário realiza a pesquisa de um alvará que não existe, verificar no Apêndice H. O sistema não permite a pesquisa de um campo nulo, ou seja, em branco e para realizar a pesquisa do cnpj é necessário utilizar a máscara com os números. O sistema permite a pesquisa de termos equivalentes com letras maiúscula, minúsculas, com acento ou sem acento.

5.3.2 Segunda Iteração

A segunda iteração de desenvolvimento consiste em desenvolver a pesquisa de um alvará por categoria e [GPS](#) e mostrar os dados do alvará. A pesquisa por categorias permitem a consulta de diversas categorias como: bares, casas noturnas, lojas comerciais, órgãos estaduais, órgãos municipais, órgãos federais, entre outras. O nome da Unipampa será utilizado no exemplo mostrado, a seguir, das telas do funcionamento da pesquisa de alvarás por categoria órgão federal e podem ser visualizadas na [Figura 46](#). O método pesquisa de alvará por categorias segue o mesmo molde da primeira iteração, a principal diferença é que ele pega a resposta do *web service* como um *vector* de soapobject retornando uma lista com os alvarás das categorias, e exibindo os alvarás em uma listview na activity, como pode-se observar na [Figura 47](#), no Apêndice H.

O método de buscar um alvará pelo [GPS](#) busca como parâmetro uma sigla, e as coordenadas de altitude e latitude são adicionadas no método onMapReady, conforme o método na [Figura 48](#). A tela do funcionamento da pesquisa de alvará por [GPS](#), utiliza o nome da Unipampa como exemplo, e pode ser visualizada na [Figura 49](#), no Apêndice H.

5.3.3 Terceira Iteração

A terceira iteração de desenvolvimento consiste em desenvolver a pesquisa para mostrar os dados dos contatos dos alvarás. O método mostrar os dados dos contatos dos alvarás, segue o mesmo molde do método da pesquisa de alvarás pela categoria pegando a resposta do *web service* como um *vector* de soapobject retornando uma lista com os dados dos contatos dos alvarás, e exibindo em uma listview. A tela do funcionamento da pesquisa dos dados dos contatos dos alvarás pode ser visualizada na [Figura 50](#), no Apêndice H. A última tela do aplicativo é uma tela de ajuda, na [Figura 51](#).

5.4 Fase de Transição

No ciclo de vida de transição do processo [OpenUP](#) as atividades a serem desenvolvidas segundo [Balduino \(2007, p.7\)](#) consiste em “realizar os testes para validar as ex-

pectativas do usuário”. A seção está organizada do seguinte modo: Na seção 5.4.1 aborda os testes com *web service* e o banco de dados MySQL. Na seção 5.4.2 aborda sobre a validação com usuários utilizando uma abordagem exploratória, através da observação do uso do sistema. Na seção 5.4.3 aborda os testes de compatibilidade que foram realizados com diferentes versões do Android. Os testes foram realizados utilizando, como exemplo, o nome da Universidade Federal do Pampa, campus Alegrete.

5.4.1 Testes com Web Service

Os testes com *web service* ocorreram juntamente com o desenvolvimento das iterações do sistema. Esses testes com *web service* e o banco de dados testam todas operações na parte do servidor e sua conexão com o banco de dados. Na [Figura 52](#) mostra como adicionar o arquivo xml, gerado ao criar o web service, no software SoapUI para realizar os testes. A [Figura 53](#) mostra o teste da inserção de dados. A [Figura 54](#) mostra o teste da atualização de dados, [Figura 55](#) mostra a exclusão de dados. A [Figura 56](#) mostra a pesquisa de dados por cnpj. A [Figura 57](#) mostra a pesquisa de dados por sigla. A [Figura 58](#) mostra a pesquisa de dados por nome fantasia. A [Figura 59](#) mostra a pesquisa do alvará por categoria. A [Figura 60](#) que mostra a pesquisa do alvará por GPS e na [Figura 61](#) mostra a pesquisa dos dados dos contatos alvará, no Apêndice I.

5.4.2 Validação com Potenciais Usuários

A validação dos protótipos funcionais foram realizadas com potenciais usuários através da observação do uso do sistema no ambiente de produção, utilizando o processo de design centrado no usuário paralelo ao processo de engenharia de software OpenUP. Os usuários utilizaram o sistema e sugeriram algumas melhorias a serem realizadas na interface do sistema. Os pontos de melhorias evidenciados pelos usuários estão na [Figura 17](#).

Figura 17 – Pontos de Melhorias sugeridos pelos usuários.

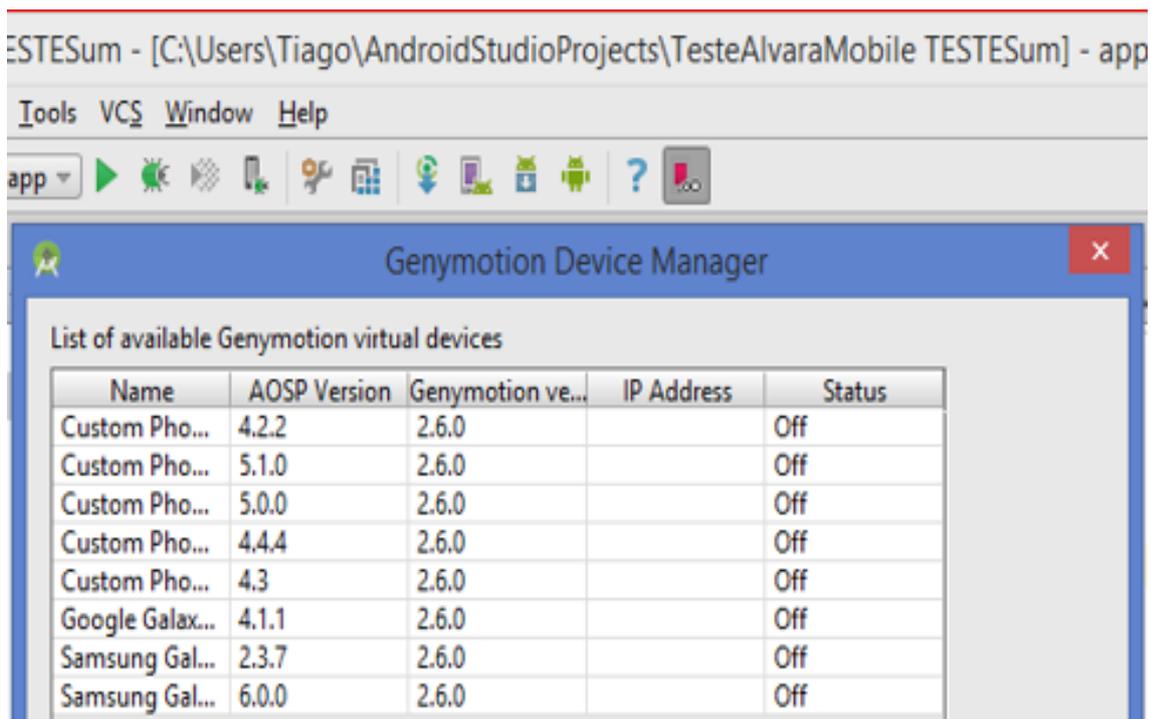
Pontos de Melhorias sugeridos	Foi feito?	(Não?) Por quê?
1 Melhorar a visibilidade da interface de ajuda ao usuário.	Sim	
2 Separar o menu de pesquisa das categorias da pesquisa por GPS.	Sim	
3 Mostrar os dados como situação, data de validade dos alvarás na pesquisa de um alvará por GPS.	Sim	
4 Melhorar a visibilidade dos dados da pesquisa por cnpj, sigla e nome, evidenciando os dados mais relevantes como a situação, validade, nome.	Não	A requisição é pertinente, mas pode ficar para uma evolução do sistema em um trabalho futuro.
5 Mostrar o nome dos alvarás nas pesquisas por categoria com um menu expandível para mostrar os dados.	Não	A requisição é pertinente, mas pode ficar para uma evolução do sistema em um trabalho futuro.

Os pontos de melhorias sugeridos pelos usuários são importantes, sob o ponto de vista da evolução do sistema porque as solicitações sugeridas pelos potenciais usuários são todas pertinentes e contribuem para melhorar a usabilidade do sistema, utilizando o processo de IHC de design centrado no usuário.

5.4.3 Teste de Compatibilidade

Os testes de compatibilidade foram realizados nas seguintes versões do Android: 2.3.7 GingerBread, 4.1.1 JellyBean, 4.2.2 JellyBean, 4.3 JellyBean, 4.4.4 KitKat, 5.0.0 Lollipop, 5.1.0 Lollipop e 6.0.0 Marshmallow. A [Figura 18](#) mostra as versões do Android configuradas para a realização dos testes. Nos testes foram testadas as funcionalidades do sistema de pesquisar um alvara por nome, sigla e cnpj, pesquisar alvará por categorias e pesquisar alvará por [GPS](#).

Figura 18 – Versões dos emuladores Android configurados para os testes.



Os testes da versão Android 2.3.7 GingerBread foram executados e funcionaram as pesquisas por nome, sigla e cnpj, as outras funcionalidades não executaram porque os componentes de interface dessa versão são obsoletos em comparação aos componentes utilizados para o desenvolvimento do aplicativo, conforme a [Figura 62](#) no Apêndice J. Os testes da versão Android 4.1.1 JellyBean funcionaram todas as funcionalidades, conforme a [Figura 63](#) no Apêndice J. Os testes da versão Android 4.2.2 JellyBean funcionaram todas as funcionalidades, conforme a [Figura 64](#) no Apêndice J. Os testes da versão Android 4.3

JellyBean funcionaram todas as funcionalidades, conforme a [Figura 65](#) no Apêndice J. Os testes da versão Android 4.4.4 KitKat funcionaram todas as funcionalidades, conforme a [Figura 66](#) no Apêndice J. Os testes da versão Android 5.0.0 Lollipop funcionaram todas as funcionalidades, conforme a [Figura 67](#) no Apêndice J. Os testes da versão Android 5.1.0 Lollipop funcionaram todas as funcionalidades, conforme a [Figura 68](#) no Apêndice J. Os testes da versão Android 6.0.0 Marshmallow estão funcionando as funcionalidades, mas ocorreu uma falha na instalação do pacote da google services no emulador, por isso não foi possível executar a funcionalidade da pesquisa dos alvarás por GPS, conforme a [Figura 69](#) no Apêndice J.

5.5 Considerações do Capítulo

Neste capítulo foram gerados os artefatos do desenvolvimento do aplicativo, utilizando o processo de engenharia de software [OpenUP](#) e o processo de Interação Humano-Computador de *design* centrado no usuário. Na fase de iniciação foram realizadas algumas etapas como: Verificar se o desenvolvimento era viável; propor uma visão geral da solução; realizar a elicitação e análise dos requisitos, baseado no modelo de cartão de histórias de usuários e entrevistas informais. Na fase de elaboração foi iniciada a primeira iteração onde foram realizadas as seguintes atividades: Detalhamento dos requisitos; validação dos requisitos com os usuários; geração do diagrama de casos de uso; definição da arquitetura; construção do diagrama de classes; construção do diagrama relacional e diagramas de sequência. Na fase de construção foram desenvolvidas as iterações do sistema. Na fase de transição foram realizados testes com o *web service*, testes com diferentes versões do android, e validações com o usuário. O desenvolvimento dessas fases foram importantes para a construção do sistema.

6 Considerações Finais

Visto que, o objetivo específico de realizar a criação de um aplicativo móvel que permita consultar a regularidade do alvará de funcionamento das organizações foi realizado. E o objetivo específico de realizar a criação de um *web service* para consultar os dados cadastrados no banco de dados de um órgão regulador e disponibilizar esses dados através do aplicativo, também foi realizado. O objetivo geral do trabalho de desenvolver uma aplicação móvel que disponibilize informações sobre a regularidade do alvará de funcionamento das organizações foi atendido.

A pesquisa sobre o estado atual para a obtenção de informações dos dados dos alvarás dos estabelecimentos foi importante para obter um entendimento sobre o funcionamento desse processo. A partir do estudo do estado atual do processo foi proposta uma solução de sistema. O estudo dos processos atual e futuro e a utilização dos cartões de histórias de usuários, junto com as entrevistas informais que foram realizadas com os potenciais usuários, empresários e representantes do órgão regulador, foram importantes para perceber a viabilidade para construção do sistema e a utilidade do sistema para os potenciais usuários, porque o aplicativo tem o intuito de ajudar os usuários na consulta dos alvarás de funcionamento de estabelecimentos cadastrados no órgão regulador, e também auxiliar o mesmo na divulgação dos estabelecimentos credenciados ao público. Na escolha da arquitetura do sistema foi possível perceber que a sua construção era viável.

Na especificação de requisitos foram utilizadas *user stories*, que logo em seguida foram convertidas em casos de uso. Os casos de uso contribuíram para complementar as *user stories*, com o intuito de destacar as funcionalidades do sistema, de um modo visual e simplificado, para o desenvolvedor, sob o ponto de vista de projeto.

Os testes com *web service* e os testes de compatibilidade mostram que as funcionalidades do sistema estão estáveis, em termos de funcionamento. A validação do sistema, utilizando a modelo de processo de IHC design centrado no usuário, com os potenciais usuários do sistema, evidenciam que o sistema necessita de evoluções na interface para melhorar a usabilidade do sistema com o usuário. A arquitetura do sistema atual acessa apenas uma base de dados MySQL através de um *web service*, mas ela é uma arquitetura que permite a expansão o sistema, em trabalhos futuros de iniciação científica, para adicionar mais *web service* que consultem outros tipos bancos de dados de diferentes órgãos reguladores como: Oracle, FirebirdSQL, PostgreSQL, entre outros.

Nos trabalhos futuros, como legado deste trabalho é possível realizar estudos de iniciação científica para melhorar a usabilidade da interface do sistema; realizar os testes com usuários em dispositivos reais para a operacionalização do sistema; integrar bancos

de dados distribuídos e heterogêneos, com o intuito de consultar diversas bases de dados de órgãos reguladores diferentes; integrar o sistema com grande quantidade de dados reais.

Referências

- APPLE. *ios*. 2015. Disponível em: <<http://www.apple.com/br/ios/what-is/>>. Acesso em: 05.11.2015. Citado na página 26.
- AXIS2, A. *Apache Axis2*. 2016. Disponível em: <<http://axis.apache.org/axis2/java/core/download.html>>. Acesso em: 05.01.2016. Citado na página 31.
- BALDUINO, R. *OpenUp*. 2007. Disponível em: <<https://eclipse.org/epf/general/OpenUP.pdf>>. Acesso em: 05.11.2015. Citado 5 vezes nas páginas 33, 43, 46, 51 e 52.
- BARBOSA, S.; SILVA, B. *Interação Humano Computador*. 2010. Citado 2 vezes nas páginas 33 e 34.
- BECK, K. *Extreme programming explained: embrace change*. [S.l.]: addison-wesley professional, 2000. Citado na página 32.
- BLACKBERRY. *company*. 2015. Disponível em: <<http://us.blackberry.com/company.html>>. Acesso em: 16.10.2015. Citado na página 26.
- BOMBEIROS, C. *Siat Mobile - Bombeiros ES*. 2015. Disponível em: <https://play.google.com/store/apps/details?id=br.gov.es.cb.gti.siat&hl=pt_BR>. Acesso em: 20.10.2015. Citado na página 40.
- BRASIL. *Lei nº 1.171 de 24 de julho*. 1996. Disponível em: <<http://www.fazenda.df.gov.br/aplicacoes/legislacao/legislacao/TelaSaidaDocumento.cfm?txtNumero=1171&txtAno=1996&txtTipo=5&txtParte=>>. Acesso em: 24.08.2015. Citado na página 21.
- CHENG, Q. et al. Gaittrack: Health monitoring of body motion from spatio-temporal parameters of simple smart phones. *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics*, 2011. Citado 3 vezes nas páginas 38, 39 e 40.
- COHN, M. *User stories applied: For agile software development*. [S.l.]: Addison-Wesley, 2004. Citado 2 vezes nas páginas 35 e 45.
- COSTA, N. P. O.; FILHO, N. F. D. Análise e avaliação funcional de sistemas operacionais móveis: vantagens e desvantagens. *Revista de Sistemas e Computação-RSC*, p. 66, 2013. Citado 3 vezes nas páginas 25, 26 e 27.
- DALLOGLIO, M. Aplicativo android para o ambiente univates virtual. 2014. Citado 3 vezes nas páginas 31, 39 e 40.
- ECLIPSE. *OpenUp*. 2009. Disponível em: <http://epf.eclipse.org/wikis/openuppt/openup_basic/guidances/roadmaps/openup_basic_roadmap,_vEruwNrEdqiM_wFaqLjNg.html>. Acesso em: 01.11.2015. Citado na página 43.
- ECLIPSE. *Eclipse Mars*. 2016. Disponível em: <<http://eclipse.org/downloads>>. Acesso em: 05.01.2016. Citado na página 31.

- ELINUX. *Arquitetura Android*. 2015. Disponível em: <http://elinux.org/Android_Architecture>. Acesso em: 20.10.2015. Citado na página 28.
- FERREIRA, M. C. et al. A proposal for a public transport ticketing solution based on customers' mobile devices. *Procedia-Social and Behavioral Sciences*, 2014. Citado 2 vezes nas páginas 38 e 40.
- FREIRE, H. Web services: A nova arquitetura da internet. *Developers Magazine*, 2002. Citado 2 vezes nas páginas 30 e 31.
- GENYMOTION. *Genymotion*. 2016. Disponível em: <<https://www.genymotion.com>>. Acesso em: 06.02.2016. Citado 2 vezes nas páginas 29 e 30.
- GONÇALVES, J. C. *Uso da plataforma android em um protótipo de aplicativo coletor de consumo de gás natural*. Curitiba, 2012. Citado na página 21.
- GOOGLE. *Google*. 2016. Disponível em: <<https://developer.android.com/studio/index.html>>. Acesso em: 16.01.2016. Citado 2 vezes nas páginas 29 e 30.
- GOOGLE, P. *Google*. 2015. Disponível em: <https://play.google.com/store?hl=pt_BR>. Acesso em: 20.10.2015. Citado 3 vezes nas páginas 24, 38 e 40.
- HP, H.-P. D. C. *webos*. 2015. Disponível em: <<http://www.hpwebos.com/us>>. Acesso em: 16.10.2015. Citado na página 27.
- IBM. *OpenUp*. 2009. Disponível em: <http://www.ibm.com/developerworks/br/rational/local/open_up/index.html>. Acesso em: 01.11.2015. Citado 3 vezes nas páginas 32, 33 e 34.
- INTEL. *Intel and Nokia Merge Software Platforms for Future Computing Devices*. 2015. Disponível em: <<http://www.intel.com/pressroom/archive/releases/2010/20100215corp.html>>. Acesso em: 16.10.2015. Citado na página 27.
- JUNIOR, C. P. N. et al. Transformações recentes na indústria de telefonia móvel: um exame de tecnologias emergentes. p. 1, 2014. Citado na página 25.
- KIELTYKA, V. *SIRASS: sistema de rastreamento de smartphone*. [S.l.], 2014. Citado na página 30.
- KIM, S. et al. Development of a mobile application, “wild flowers of bukhansan national park (version 1.0)”, for identification of plants in bukhansan national park. *Journal of Korean Nature*, 2011. Citado 3 vezes nas páginas 38, 39 e 40.
- KSOAP2. *Ksoap2*. 2016. Disponível em: <<http://www.java2s.com/Code/Jar/k/Downloadksoap2android252jar.htm>>. Acesso em: 05.01.2016. Citado na página 31.
- LABDARIU, V. *TRAFFI*. 2015. Disponível em: <<https://play.google.com/store/apps/details?id=com.trafi.android.tr>>. Acesso em: 20.10.2015. Citado na página 40.
- LARMAN, C. *Agile project management with Scrum*. [S.l.]: Bookman Editor, 2002. Citado na página 32.
- LECHETA, R. R. *Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK*. [S.l.]: Editora Novatec, 2009. 21 p. Citado 6 vezes nas páginas 25, 26, 28, 29, 30 e 32.

- LECHETA, R. R. *Android Essencial*. [S.l.]: Editora Novatec, 2016. Citado na página 27.
- MILANI, A. *Programando para iPhone e iPad: Aprenda a construir aplicativos para o iOS*. [S.l.]: Editora Novatec, 2012. Citado na página 26.
- MONACO, T.; CARMO, R. M. D. *Desenvolvendo aplicações para Windows PhoneS*. [S.l.]: Editora Brasport, 2012. Citado na página 27.
- MYSQL. *MySQL*. 2016. Disponível em: <<http://dev.mysql.com/downloads/connector>>. Acesso em: 05.01.2016. Citado na página 31.
- PERUMAL, T. et al. Development of an embedded smart home management scheme. *International Journal of Smart Home*, 2013. Citado na página 32.
- POCATILU, P. Developing mobile learning applications for android using web services. *Informatica Economica*, 2010. Citado 2 vezes nas páginas 39 e 40.
- PRESSMAN, R. S. *Engenharia de software*. [S.l.]: McGraw Hill Brasil, 2011. Citado na página 35.
- REGO, G. B. *UserStories*. 2012. Disponível em: <<https://code.google.com/p/extreme-programming-mds/wiki/UserStories>>. Acesso em: 05.11.2015. Citado na página 35.
- RICALDI, T.; TOLFO, C. Proposta de aplicação móvel para a consulta de alvarás de funcionamento. *VII SIEPE - Salão Internacional de Ensino, Pesquisa e Extensão*, 2015. Citado 2 vezes nas páginas 32 e 40.
- SANTOS, J. d. Debitar: gerenciamento de débitos em android com sincronização web/jsf através de web service. 2013. Citado na página 29.
- SCHWABER, K. *Agile project management with Scrum*. [S.l.]: Microsoft press, 2004. Citado na página 32.
- SETIAWAN, E. I. et al. Shortest path problem for public transportation using gps and map service. *Procedia-Social and Behavioral Sciences*, 2012. Citado 2 vezes nas páginas 38 e 40.
- SHANMUGAPRIYA, M.; TAMILARASI, A. Designing an m-learning application for a ubiquitous learning environment in the android based mobile devices using web services. *Indian Journal of Computer Science and Engineering*, 2011. Citado 2 vezes nas páginas 39 e 40.
- SMARTBEAR. *SoapUI*. 2016. Disponível em: <<http://www.soapui.org/downloads/latest-release.html>>. Acesso em: 05.01.2016. Citado na página 31.
- SOMMERVILLE, I. *Engenharia de Software*. [S.l.]: Pearson, 2011. Citado 4 vezes nas páginas 35, 36, 45 e 47.
- TOMCAT. *Tomcat*. 2016. Disponível em: <<http://www.tomcat.apache.org>>. Acesso em: 05.01.2016. Citado na página 31.
- W3SCHOOLS. *Services*. 2015. Disponível em: <http://www.w3schools.com/xml/xml_services.asp>. Acesso em: 26.10.2015. Citado na página 32.

XAMPP. *Xampp*. 2016. Disponível em: <<http://www.apachefriends.org>>. Acesso em: 05.01.2016. Citado na página 31.

Apêndices

APÊNDICE A – Apêndice A

Figura 19 – Cartões de estórias do usuário 1.

Cartão 1 de Estória de Usuário			
Usuário 1: COMO usuário do aplicativo, QUERO buscar os dados de um alvará pelo nome fantasia da empresa PARA poder realizar a consulta.			
Cartão 2 de Estória de Usuário			
Usuário 1: COMO usuário do aplicativo, QUERO buscar os dados de um alvará pelo CNPJ da empresa PARA poder realizar a consulta.			
Cartão 3 de Estória de Usuário			
Usuário 1: COMO usuário do aplicativo, QUERO buscar os dados de um alvará pela localização do GPS PARA poder realizar a consulta.			

Figura 20 – Cartões de estórias do usuário 2.

Cartão 4 de Estória de Usuário	Cartão 7 de Estória de Usuário
Usuário 2 : COMO usuário do aplicativo, GOSTARIA de identificar a lotação permitida de um estabelecimento PARA verificar a capacidade máxima de pessoas.	Usuário 2 : COMO usuário do aplicativo, GOSTARIA de realizar busca por SIGLA PARA consultar o alvará.
Cartão 5 de Estória de Usuário	Cartão 8 de Estória de Usuário
Usuário 2 : COMO usuário do aplicativo, GOSTARIA de pesquisar estabelecimentos por categoria de serviço/produto que oferecem PARA consultar o alvará.	Usuário 2 : COMO usuário do aplicativo, GOSTARIA que o mecanismo de busca não exigisse que eu digitasse o termo de busca de forma exata, considerando equivalentes termos como: Unipampa, UNIPAMPA, unipampa PARA facilitar a busca.
Cartão 6 de Estória de Usuário	
Usuário 2 : COMO usuário do aplicativo, GOSTARIA que o mecanismo de busca me auxiliasse com o recurso de auto sugestão PARA facilitar a busca.	

Figura 21 – Cartão de estória do usuário 3.

Cartão 9 de Estória de Usuário
Usuário 3 : COMO um fiscal do órgão regulador, GOSTARIA que o aplicativo disponibilizasse o horário de funcionamento de uma empresa nos dados do alvará PARA verificar o horário.

Figura 22 – Cartões de estórias do usuário 4.

Cartão 10 de Estória de Usuário
Usuário 4 : COMO um fiscal do órgão regulador, GOSTARIA que o aplicativo exibisse a data de validade de um alvará PARA verificar os dados.
Cartão 11 de Estória de Usuário
Usuário 4 : COMO um fiscal do órgão regulador, GOSTARIA que o aplicativo disponibilizasse todos os dados referentes ao alvará de funcionamento como: Razão Social, CNPJ, Nome fantasia, Ramo de atividade, endereço, Início da atividade, horário de funcionamento, data de validade, número de cadastro do alvará PARA verificar os dados.

Figura 23 – Cartão de estória do usuário 5.

Cartão 12 de Estória de Usuário
Usuário 5 : COMO um empresário, GOSTARIA que o aplicativo mostrasse todos os dados referentes a um alvará PARA consultar os dados do alvará da empresa.

Figura 24 – Cartão de estória do usuário 6.

Cartão 13 de Estória de Usuário
Usuário 6 : COMO um empresário, GOSTARIA que o aplicativo mostrasse o endereço e o horário de funcionamento do meu estabelecimento nos dados do alvará PARA as pessoas consultar.

APÊNDICE B – Apêndice B

Tabela 3 – UC01: Pesquisar alvará de uma organização pelo nome fantasia.

UC01- Pesquisar alvará de uma organização pelo nome fantasia.

Atores

Usuário e sistema

Pré-condição

1. O sistema deverá estar instalado no dispositivo móvel e o usuário deverá ter acesso a internet.

Fluxo Principal

1. Usuário: Acessa o aplicativo de consulta de alvarás, via touch screen.

2. Sistema: Mostra a opção de pesquisa do aplicativo na tela principal.

3. Usuário: Usuário seleciona a pesquisa e digita o nome fantasia do estabelecimento na opção de pesquisar (O recurso de auto sugestão deve estar disponível).

4. Sistema: Mostra os dados do alvará do estabelecimento regular no UC04.

Fluxos Alternativos

4 a) Sistema: Caso o alvará pesquisado não existir, mostra que o alvará do estabelecimento não foi encontrado no sistema.

b) Usuário: Visualiza a tela de alvará não encontrado.

Requisitos Não Funcionais

RNF (Requisito de Desempenho)- Ao pesquisar o alvará de um estabelecimento, os dados gerais do alvará devem aparecer em menos de 3 segundos.

Tabela 4 – UC02: Pesquisar alvará de uma organização pelo CNPJ.

UC02: Pesquisar alvará de uma organização pelo CNPJ.

Atores

Usuário e sistema

Pré-condição

1. O sistema deverá estar instalado no dispositivo móvel e o usuário deverá ter acesso a internet.

Fluxo Principal

1. Usuário: Acessa o aplicativo de consulta de alvarás, via touch screen.

2. Sistema: Mostra a opção de pesquisa do aplicativo na tela principal.

3. Usuário: Usuário seleciona a pesquisa e digita o CNPJ do estabelecimento na opção de pesquisar (O recurso de auto sugestão deve estar disponível).

4. Sistema: Mostra os dados do alvará do estabelecimento regular no UC04.

Fluxos Alternativos

4 a) Sistema: Caso o alvará pesquisado não existir, mostra que o alvará do estabelecimento não foi encontrado no sistema.

b) Usuário: Visualiza a tela de alvará não encontrado.

Requisitos Não Funcionais

RNF (Requisito de Usabilidade)- A interface deve ser minimalista para pesquisar o nome, CNPJ e Sigla no mesmo campo de pesquisa.

Tabela 5 – UC03: Pesquisar alvará de uma organização pela sigla.

UC03: Pesquisar alvará de uma organização pela sigla.**Atores**

Usuário e sistema

Pré-condição

1. O sistema deverá estar instalado no dispositivo móvel e o usuário deverá ter acesso a internet.

Fluxo Principal

1. Usuário: Acessa o aplicativo de consulta de alvarás, via touch screen.
2. Sistema: Mostra a opção de pesquisa do aplicativo na tela principal.
3. Usuário: Usuário seleciona a pesquisa e digita a sigla do estabelecimento na opção de pesquisar (O recurso de auto sugestão deve estar disponível).
4. Sistema: Mostra os dados do alvará do estabelecimento regular no UC04.

Fluxos Alternativos

4 a) Sistema: Caso o alvará pesquisado pela sigla não existir, mostra que o alvará do estabelecimento não foi encontrado no sistema.

b) Usuário: Visualiza a tela de alvará não encontrado.

Requisitos Não Funcionais

RNF (Requisito de usabilidade)- O mecanismo de busca deve considerar termos equivalentes.

Tabela 6 – UC04: Mostrar os dados do alvará da organização.

UC04: Mostrar dados do alvará da organização.**Atores**

Usuário e sistema

Pré-condição

1. O sistema deverá estar instalado no dispositivo móvel e o usuário deverá ter acesso a internet.
2. O usuário deve executar antes o RF01 ou RF02 ou RF03 ou RF05 ou RF06.

Fluxo Principal

1. Usuário: Visualiza a tela dos dados gerais do alvará.
2. Sistema: Exibe um relatório com os dados gerais do alvará como: Status, razão social, CNPJ, nome fantasia, ramo de atividade, endereço, horário de funcionamento, data de validade, número de cadastro do alvará, lotação máxima.

Requisitos Não Funcionais

RNF (Requisito de interoperabilidade)- O sistema deverá acessar um banco de dados externo à aplicação Android para buscar os dados.

Tabela 7 – UC05: Pesquisar alvará de uma organização pela categoria de serviço.

UC05: Pesquisar alvará de uma organização por categoria de serviço.**Atores**

Usuário e sistema

Pré-condição

1. O sistema deverá estar instalado no dispositivo móvel e o usuário deverá ter acesso a internet.

Fluxo Principal

1. Usuário: Acessa o aplicativo de consulta de alvarás, via touch screen.
2. Sistema: Mostra o menu do aplicativo na tela principal.
3. Usuário: Seleciona o menu e escolhe uma das categorias.
4. Sistema: Exibe uma lista de alvarás com os dados do UC 04 dos estabelecimentos da categoria escolhida.

Requisitos Não Funcionais

RNF (Requisito de usabilidade)- A interface deve ser intuitiva e simular uma interface em Android para prover uma comunicação com o usuário.

Tabela 8 – UC06: Pesquisar alvará de uma organização por GPS.

UC06: Pesquisar alvará de uma organização por GPS.**Atores**

Usuário e sistema

Pré-condição

1. O sistema deverá estar instalado no dispositivo móvel e o usuário deverá ter acesso a internet.

Fluxo Principal

1. Usuário: Acessa o aplicativo de consulta de alvarás, via touch screen.
2. Sistema: Mostra as opções de pesquisa do aplicativo na tela principal.
3. Usuário: Seleciona a pesquisa por GPS.
4. Sistema: Exibe um mapa com os estabelecimento de alvarás regulares.
5. Usuário: Escolhe um estabelecimento desejado no mapa.
6. Sistema: Mostra os dados do alvará do estabelecimento regular. Os dados exibidos são: Nome, Situação, Validade.

Requisitos Não FuncionaisRNF (Requisito de disponibilidade)- O sistema não pode ficar indisponível, sem conexão a internet.

Tabela 9 – UC07: Mostrar os dados dos contatos dos alvarás.

UC07: Mostrar os dados dos contatos dos alvarás.**Atores**

Usuário e sistema

Pré-condição

1. O sistema deverá estar instalado no dispositivo móvel e o usuário deverá ter acesso a internet.

Fluxo Principal

1. Usuário: Visualiza o menu lateral da direita do sistema e clica em contatos.
2. Sistema: Exibe um relatório com os dados dos contatos do alvará como: Nome, Descrição, Telefone, Responsável, Email.

Requisitos Não FuncionaisRNF (Requisito de interoperabilidade)- Ao pesquisar os contatos deve aparecer uma lista em menos de 3 segundos.

APÊNDICE C – Apêndice C

Figura 25 – Tela Principal.

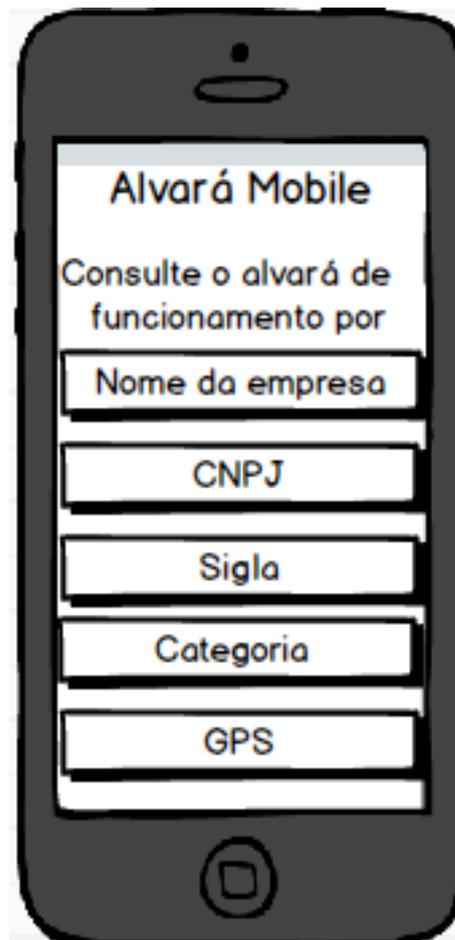


Figura 26 – Pesquisa por Nome Fantasia da Organização.



APÊNDICE D – Apêndice D

Figura 27 – Tela Principal e Pesquisa por CNPJ, Nome ou Sigla.



Figura 28 – Pesquisa por GPS.



Figura 29 – Pesquisa por Categorias.



APÊNDICE E – Apêndice E

Figura 30 – Diagrama de sequência Pesquisar alvará por CNPJ, nome fantasia e sigla.

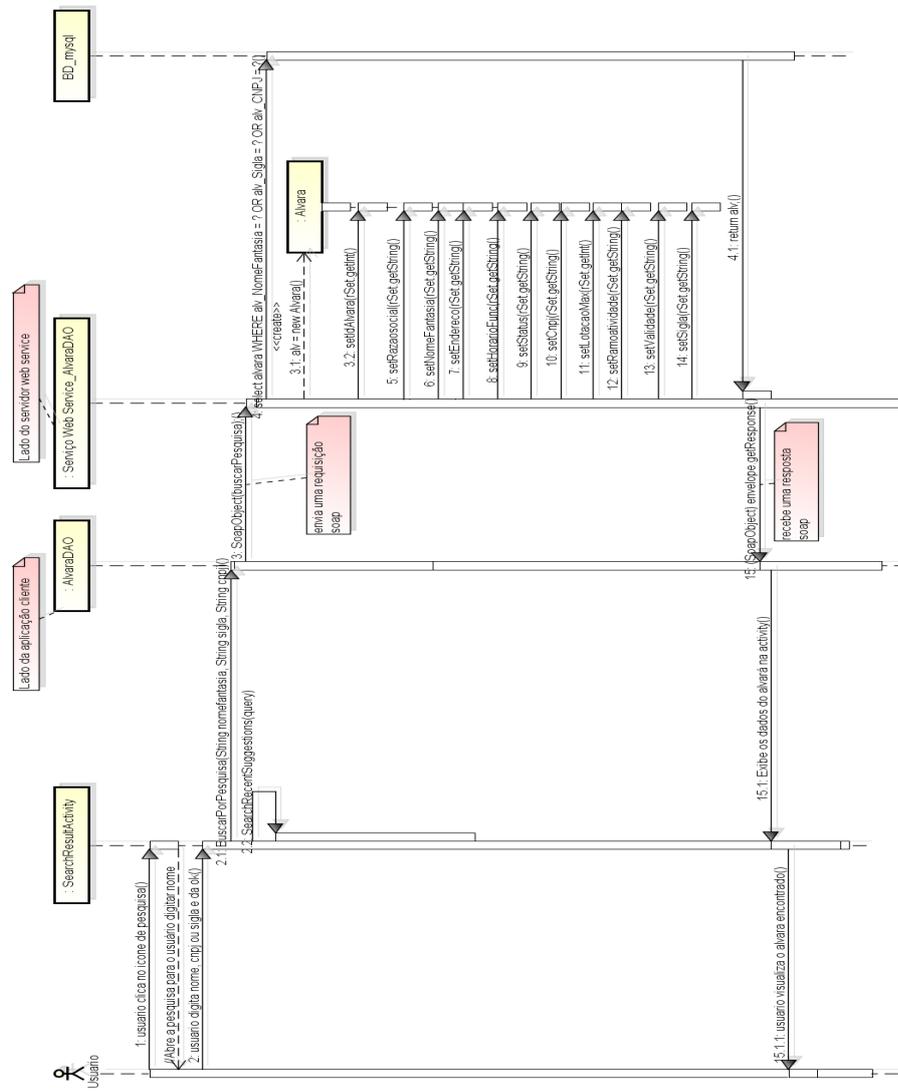


Figura 31 – Diagrama de sequência Pesquisar alvará por categoria.

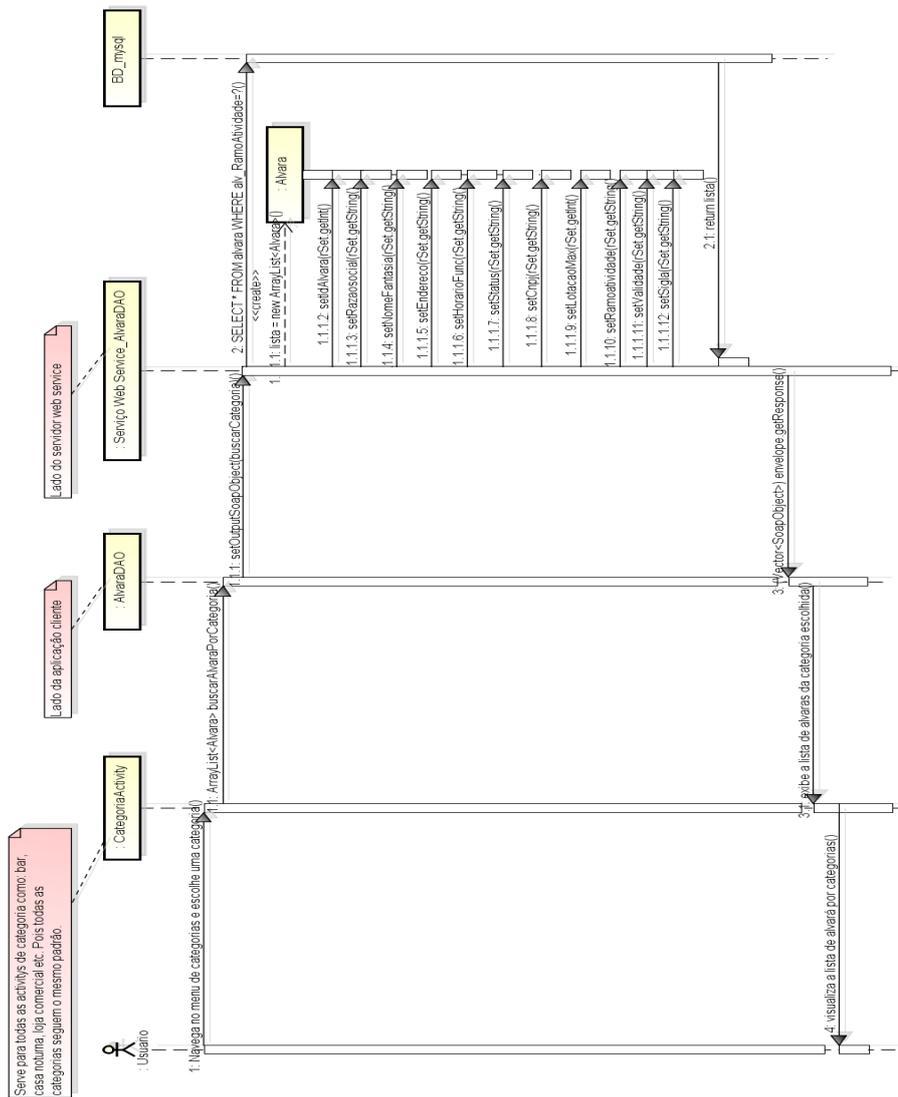


Figura 32 – Diagrama de seqüência Pesquisar alvará por GPS.

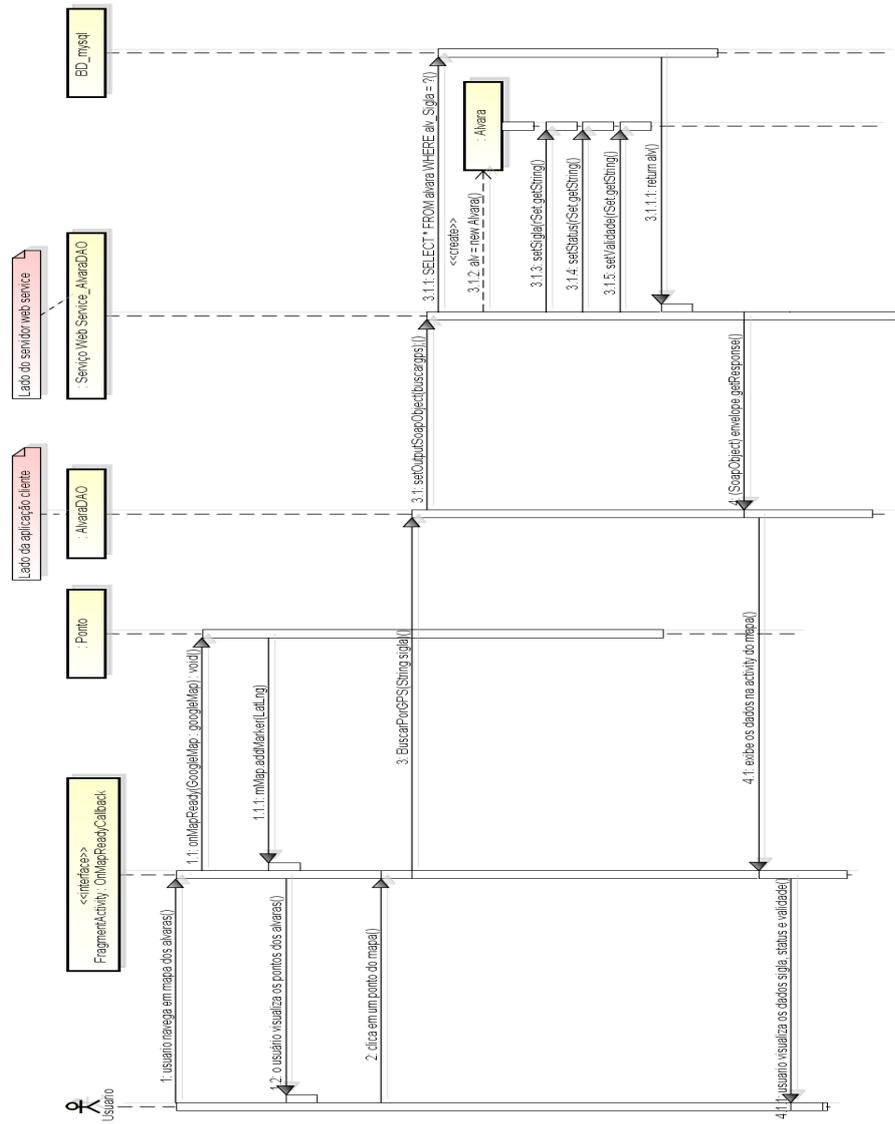
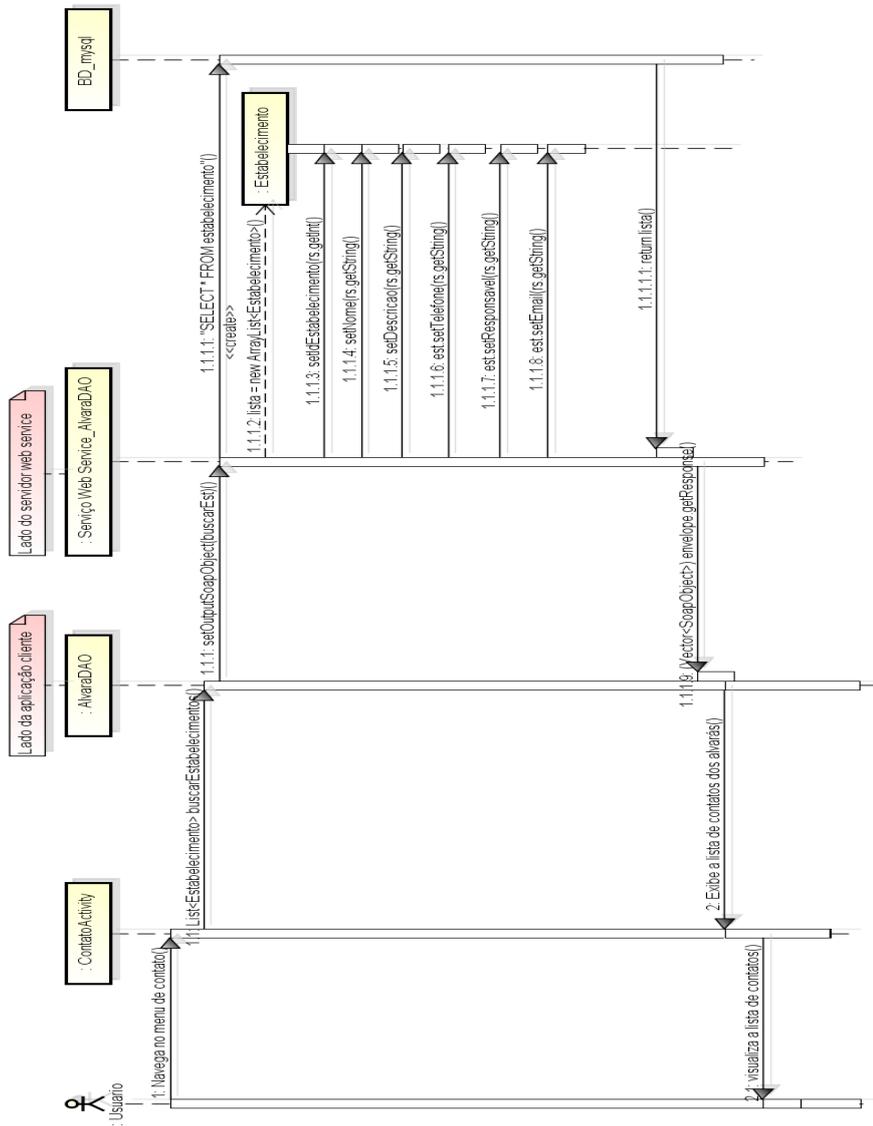
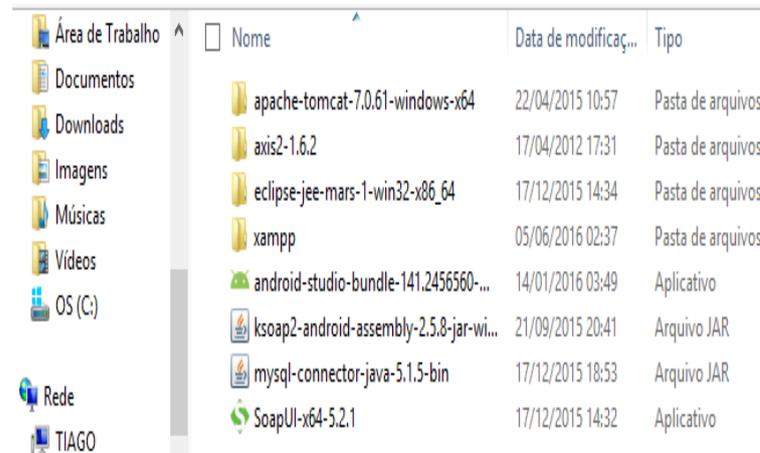


Figura 33 – Diagrama de seqüência Mostrar dados do contato do alvará.



APÊNDICE F – Apêndice F

Figura 34 – Programas necessários para o desenvolvimento.



The image shows a Windows File Explorer window with a list of files and folders. The left sidebar shows the navigation pane with 'Área de Trabalho' selected. The main pane displays a table of files with columns for 'Nome', 'Data de modifícaç...', and 'Tipo'.

Nome	Data de modifícaç...	Tipo
apache-tomcat-7.0.61-windows-x64	22/04/2015 10:57	Pasta de arquivos
axis2-1.6.2	17/04/2012 17:31	Pasta de arquivos
eclipse-jee-mars-1-win32-x86_64	17/12/2015 14:34	Pasta de arquivos
xampp	05/06/2016 02:37	Pasta de arquivos
android-studio-bundle-141.2456560-...	14/01/2016 03:49	Aplicativo
ksoap2-android-assembly-2.5.8-jar-wi...	21/09/2015 20:41	Arquivo JAR
mysql-connector-java-5.1.5-bin	17/12/2015 18:53	Arquivo JAR
SoapUI-x64-5.2.1	17/12/2015 14:32	Aplicativo

APÊNDICE G – Apêndice G

Figura 35 – Criação do Web Service AlvaraDAO passo 1.

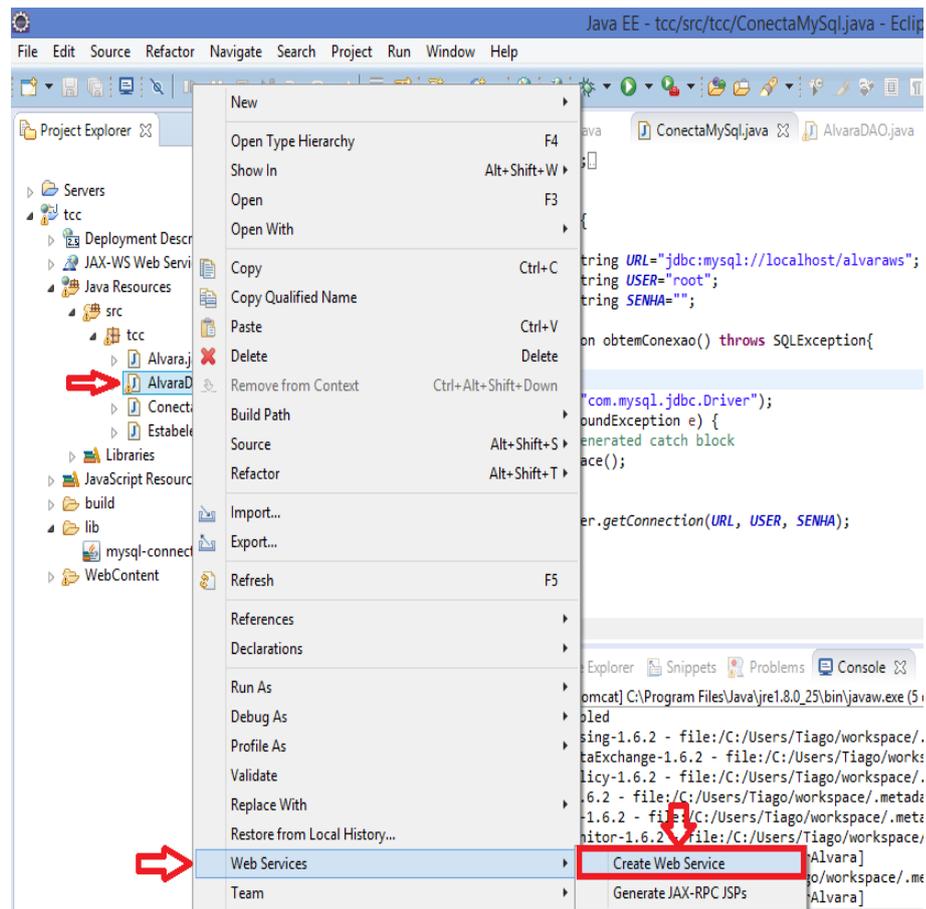


Figura 36 – Criação do Web Service AlvaraDAO passo 2.

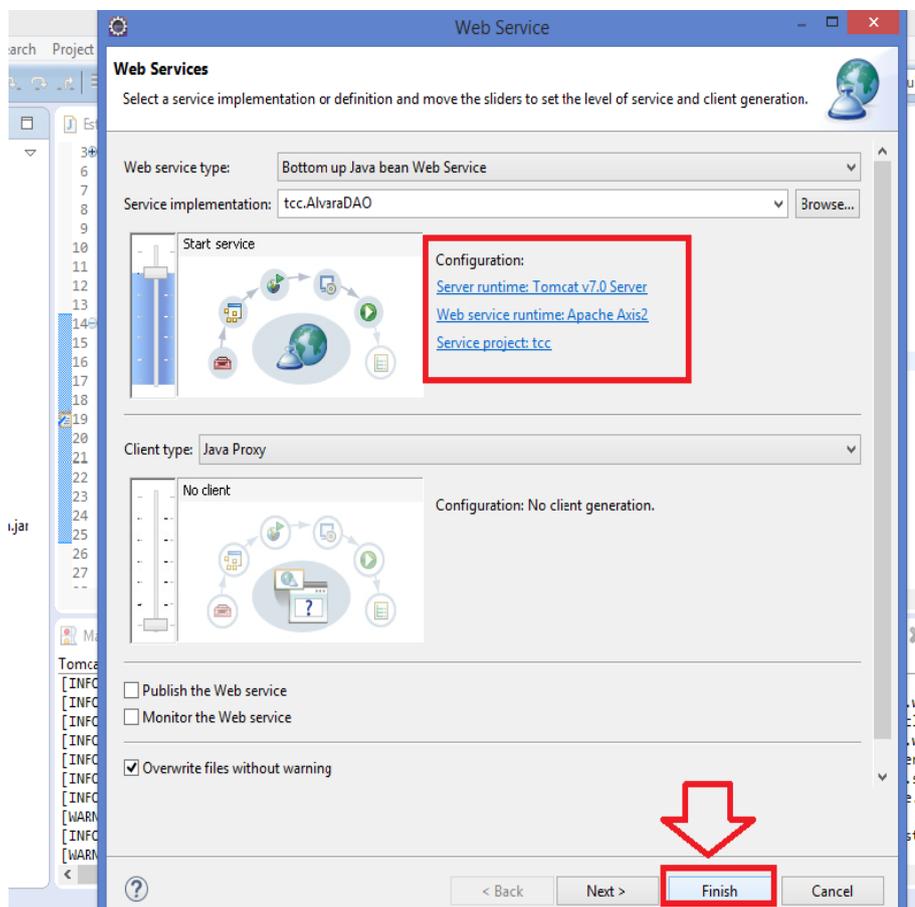


Figura 37 – Criação do Web Service AlvaraDAO passo 3.



Welcome!

Welcome to the new generation of Axis. If you can see this page you have successfully deployed the Axis2 Web Application. However, to ensure that Axis2 is prop

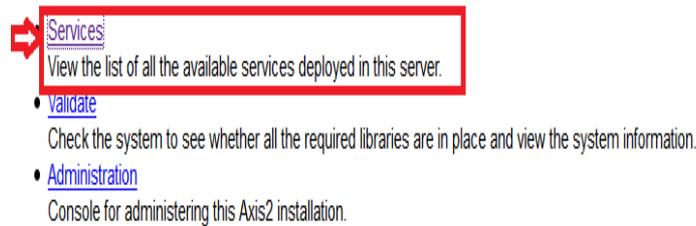


Figura 38 – Criação do Web Service AlvaraDAO passo 4.

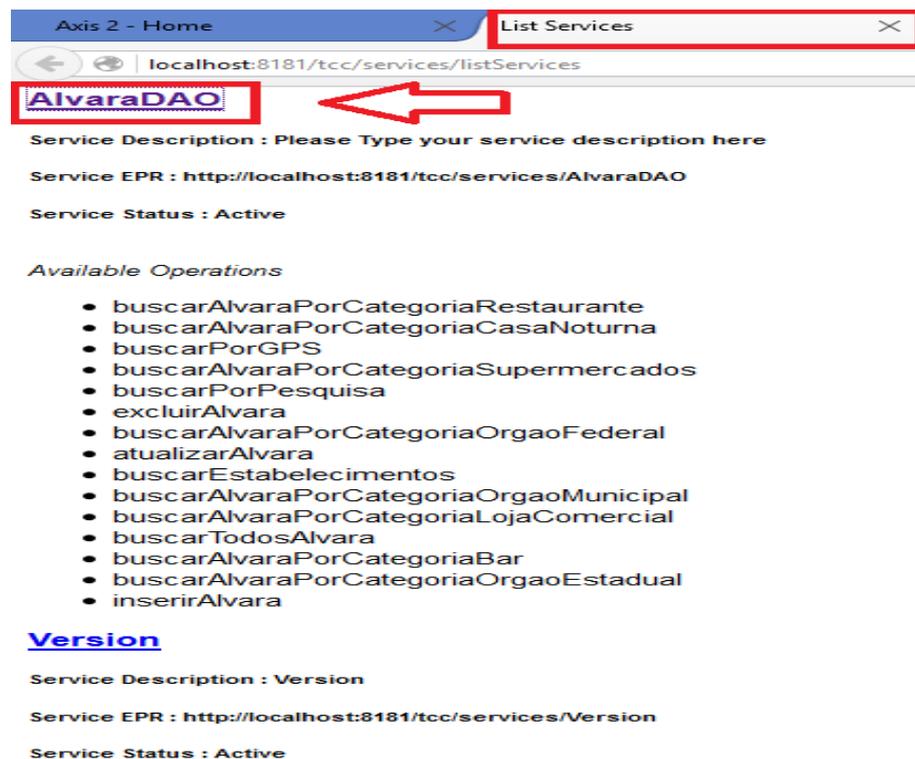


Figura 39 – Criação do Web Service AlvaraDAO Geração do XML passo 5.

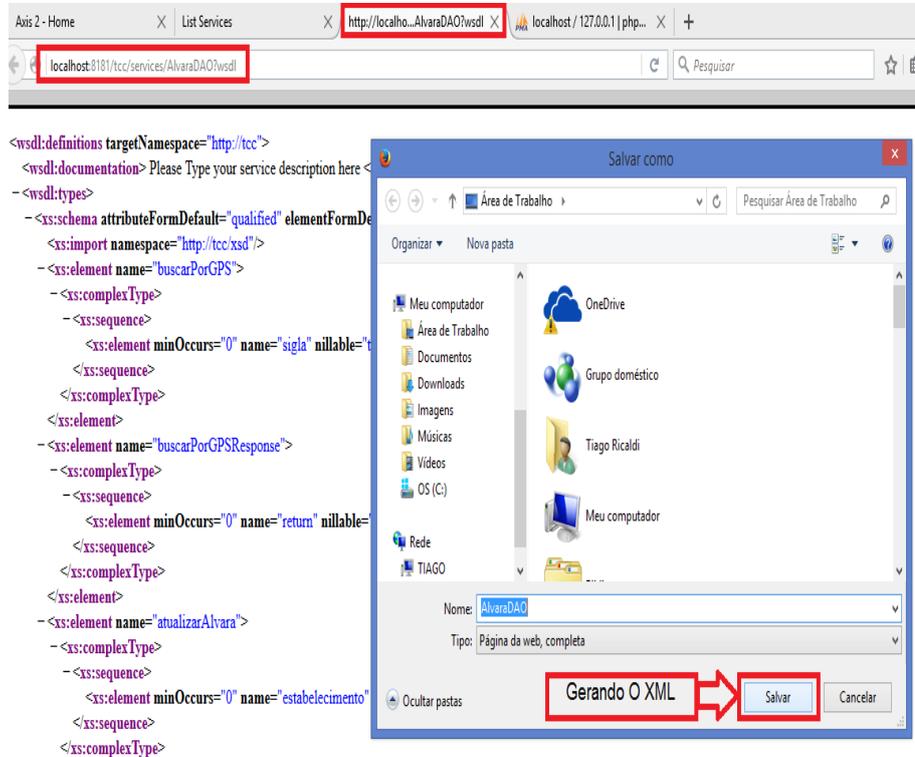
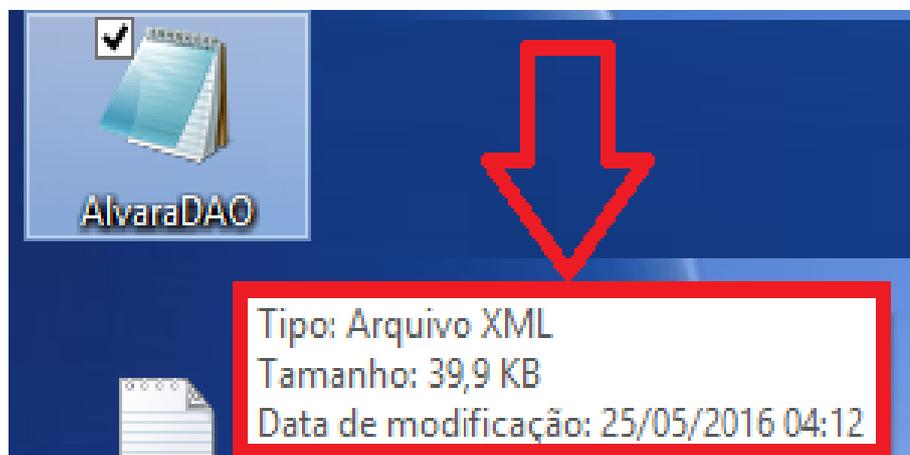


Figura 40 – Criação do Web Service AlvaraDAO XML passo 6.



APÊNDICE H – Apêndice H

Figura 41 – Classe AlvaraDAO e visão geral do projeto.

The screenshot displays the Android Studio IDE with the following components:

- Project Structure (Left):** Shows the hierarchy of the project, including the 'app' folder, 'src/main/java/com/example/...' package structure, and 'res' resources.
- Code Editor (Center):** Contains the source code for `AlvaraDAO.java`.


```

import com.example.tiago.testealvaramobile.model.Alvara;
import com.example.tiago.testealvaramobile.model.Estabelecimento;

import java.util.ArrayList;
import java.util.Vector;
import org.ksoap2.SoapEnvelope;
import org.ksoap2.serialization.SoapObject;
import org.ksoap2.serialization.SoapSerializationEnvelope;
import org.ksoap2.transport.HttpTransportSE;

public class AlvaraDAO {

    private static final String URL = "http://192.168.56.1:8181/tcc/services/AlvaraDAO?wsdl";
    private static final String NAMESPACE = "http://tcc";

    private static final String BUSCAPESQUIZA = "buscarPorPesquisa";
    private static final String BUSCAPORGPS = "buscarPorGPS";
    private static final String BUSCACATEGORIABAR = "buscarAlvaraPorCategoriaBar";
    private static final String BUSCACATEGORIACASANOTURNA = "buscarAlvaraPorCategoriaCasaNoturna";
    private static final String BUSCACATEGORIALOJACOMERCIAL = "buscarAlvaraPorCategoriaLojaComercial";
    private static final String BUSCACATEGORIAORGAOESTADUAL = "buscarAlvaraPorCategoriaOrgaoEstadual";
    private static final String BUSCACATEGORIAORGAOFEDERAL = "buscarAlvaraPorCategoriaOrgaoFederal";
    private static final String BUSCACATEGORIAORGAOMUNICIPAL = "buscarAlvaraPorCategoriaOrgaoMunicipal";
    private static final String BUSCACATEGORIARESTAURANTE = "buscarAlvaraPorCategoriaRestaurante";
    private static final String BUSCACATEGORIASUPERMERCADOS = "buscarAlvaraPorCategoriaSupermercados";
    private static final String BUSCATODOSALVARA = "buscarTodosAlvara";
    private static final String BUSCATODOESTABELECIMENTO = "buscarEstabelecimentos";

```
- Annotations (Right):**
 - A red box highlights the imports from `org.ksoap2` with the text "Import da biblioteca ksoap2".
 - An arrow points to the `URL` and `NAMESPACE` strings with the text "Local onde está web service.".
 - An arrow points to the `NAMESPACE` string with the text "Nome pacote está o web service".
 - A red box highlights the list of constants with the text "Constantes para cada método".

Figura 42 – Método buscar alvará por Pesquisa

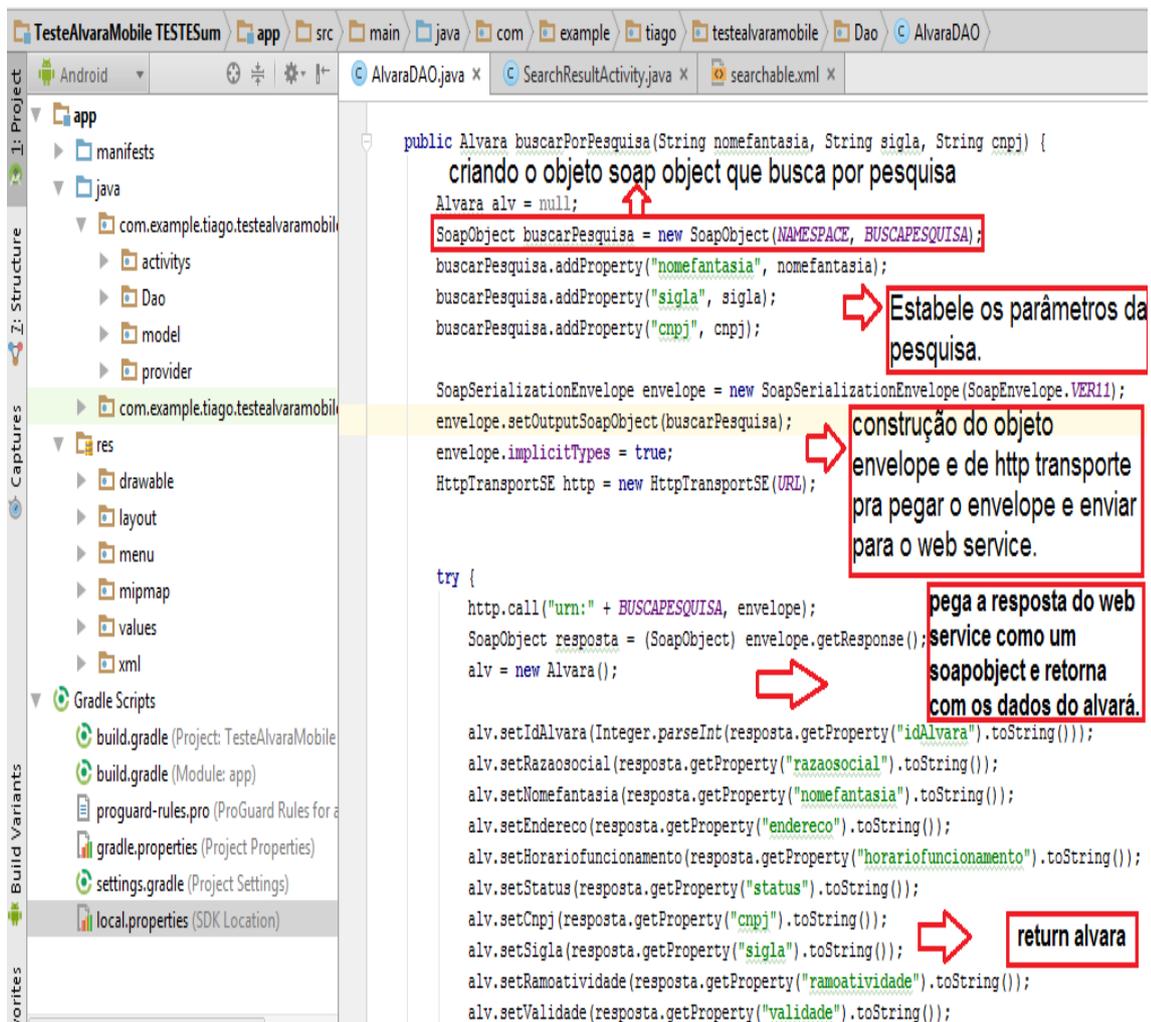


Figura 43 – Activity SearchResultActivity

```
public class SearchResultActivity extends AppCompatActivity {  
  
    Alvara alv = null;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_search_result);  
  
        Intent myintent = getIntent();  
        try {  
  
            if (Intent.ACTION_SEARCH.equalsIgnoreCase(myintent.getAction())) {  
                String query = myintent.getStringExtra(SearchManager.QUERY.toLowerCase());  
                AlvaraDAO estabelecimentoDAO1 = new AlvaraDAO();  
                alv = estabelecimentoDAO1.buscarPorPesquisa(query, query, query);  
  
                SearchRecentSuggestions searchRecentSuggestions = new SearchRecentSuggestions(this, Searcha  
                searchRecentSuggestions.saveRecentQuery(query, null);  
  
                TextView textview = (TextView) findViewById(R.id.textView_search);  
                textview.setTextSize(18);  
                textview.setText(alv.toString());  
            }  
        }  
    }  
}
```

Verifica o campo digitado ignorando se é letra maiuscula ou minuscula ou com acento ou sem e realiza a pesquisa com termos equivalentes.

chamando o método

Cria as sugestões recentes

Exibe o alvará numa textview

Figura 44 – Telas de Funcionamento da funcionalidade Buscar por cnpj, nome fantasia e sigla

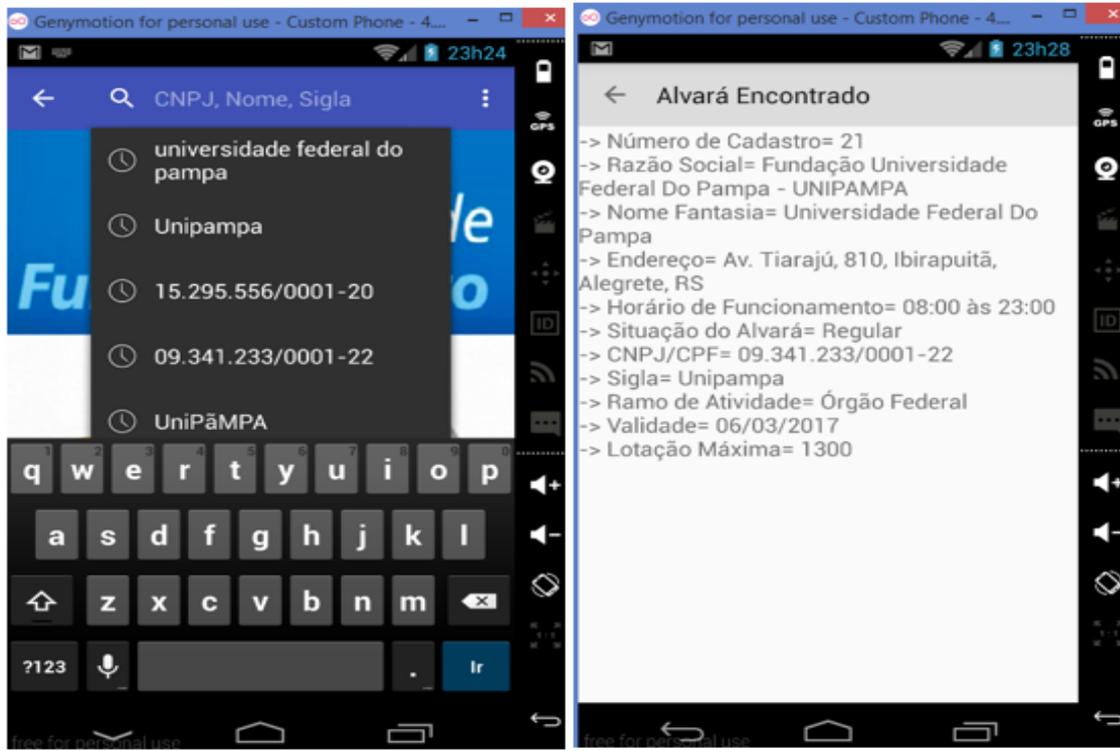


Figura 45 – Telas de Funcionamento de Alvará não encontrado

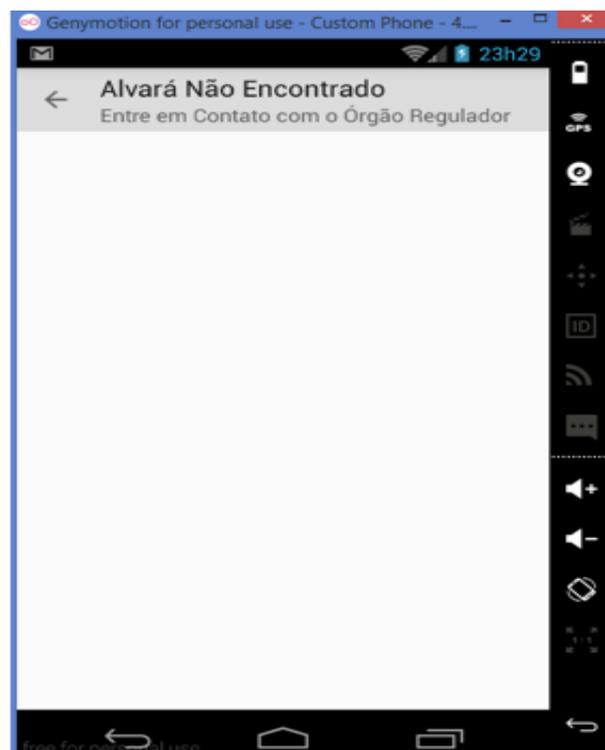


Figura 46 – Telas de Funcionamento Pesquisar por categoria órgão federal

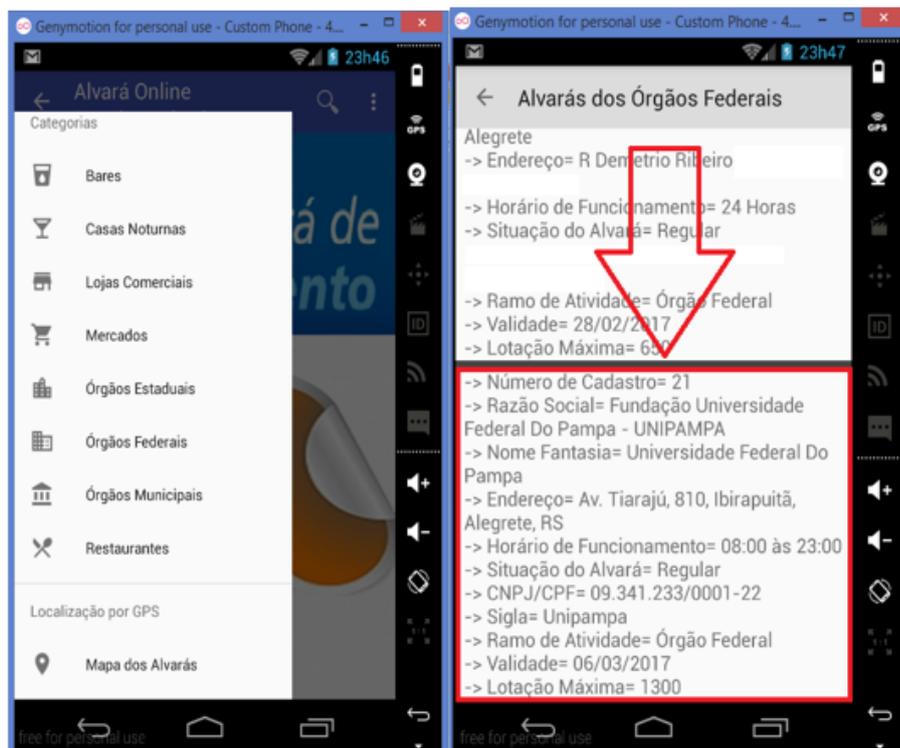


Figura 47 – Método buscar alvará por Categoria

```
try {
    http.call("urn:" + BUSCACATEGORIAORGAOFEDERAL, envelope);

    Vector<SoapObject> resposta = (Vector<SoapObject>) envelope.getResponse();

    for (SoapObject aux : resposta) {  Pega a resposta do web service como um vector de soapobject

        Alvara alv = new Alvara();

        alv.setIdAlvara(Integer.parseInt(aux.getProperty("idAlvara").toString()));
        alv.setRazaosocial(aux.getProperty("razaosocial").toString());
        alv.setNomefantasia(aux.getProperty("nomefantasia").toString());
        alv.setEndereco(aux.getProperty("endereco").toString());
        alv.setHorariofuncionamento(aux.getProperty("horariofuncionamento").toString());
        alv.setStatus(aux.getProperty("status").toString());
        alv.setCnpj(aux.getProperty("cnpj").toString());
        alv.setSigla(aux.getProperty("sigla").toString());
        alv.setRamoatividade(aux.getProperty("ramoatividade").toString());
        alv.setValidade(aux.getProperty("validade").toString());
        alv.setLotacaomaxima(Integer.parseInt(aux.getProperty("lotacaomaxima").toString()));

        lista.add(alv);  Adiciona os alvarás na lista da categoria
    }

} catch (Exception e) {
    e.printStackTrace();
    return null;
}
return lista;  Return a lista
```

Figura 48 – Método buscar alvará por GPS

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    AlvaraDAO estabelecimentoDAO1 = new AlvaraDAO();

    Alvara alv1 = estabelecimentoDAO1.buscarPorGPS("Unipampa");
    LatLng unipampa= new LatLng(-29.7896, -55.7686);
    mMap.addMarker(new MarkerOptions().position(unipampa).title(alv1.getSigla() + " Situação:" + alv1.getStatus()).snippet("Validade:" + alv1.ge
    mMap.moveCamera(CameraUpdateFactory.newLatLng(unipampa));
```

Busca pela sigla os dados

Adiciona a Latitude e altitude

Chama os atributos que deseja exibir no Ponto ou Overlay

Figura 49 – Tela de Funcionamento Buscar Alvará por GPS

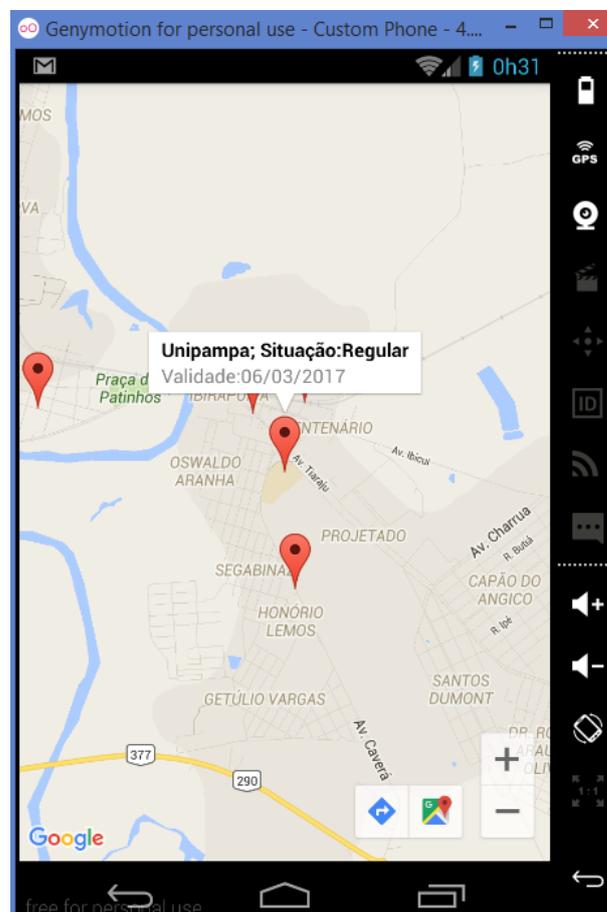


Figura 50 – Tela de Funcionamento Mostrar dados dos contatos do Alvará

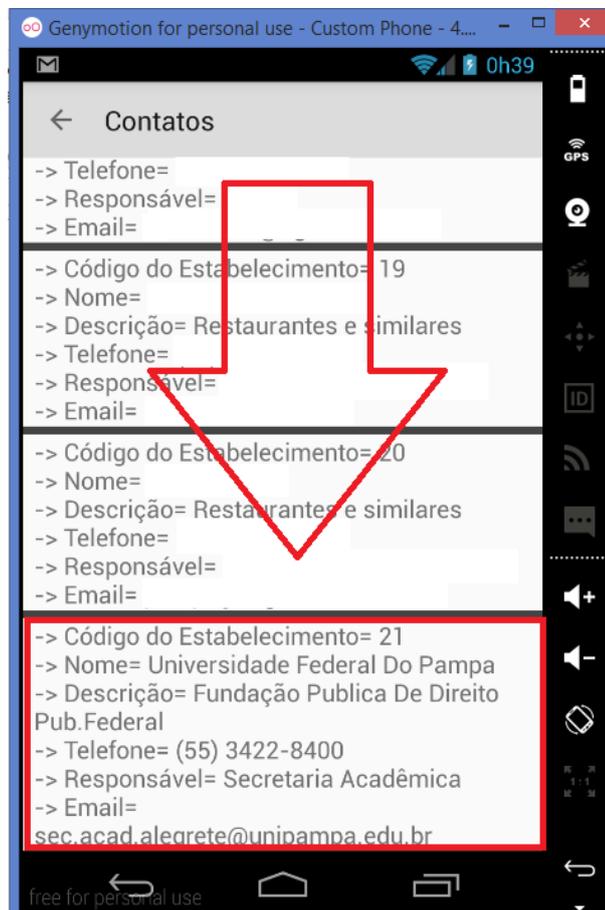
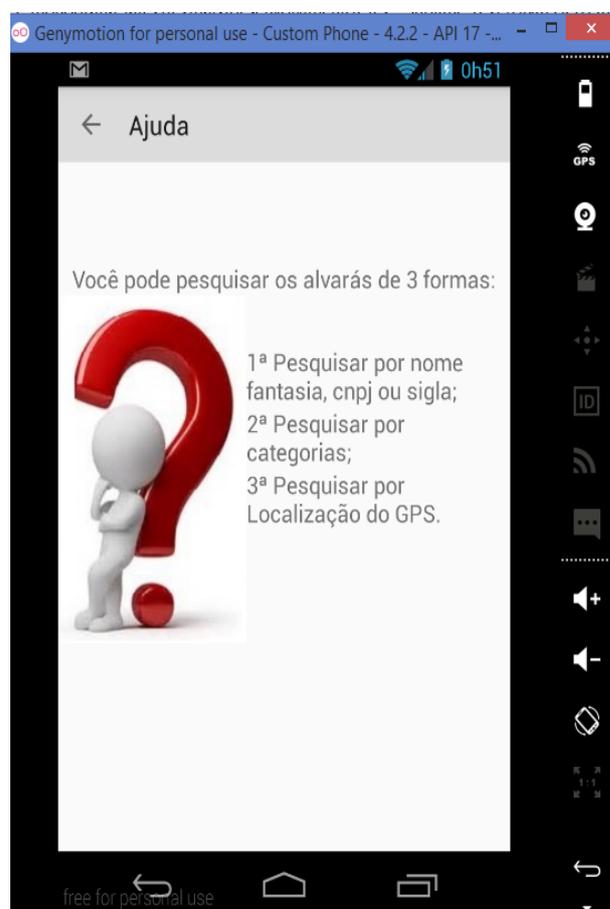


Figura 51 – Tela de ajuda



APÊNDICE I – Apêndice I

Figura 52 – Adicionando o xml no software SoapUI.

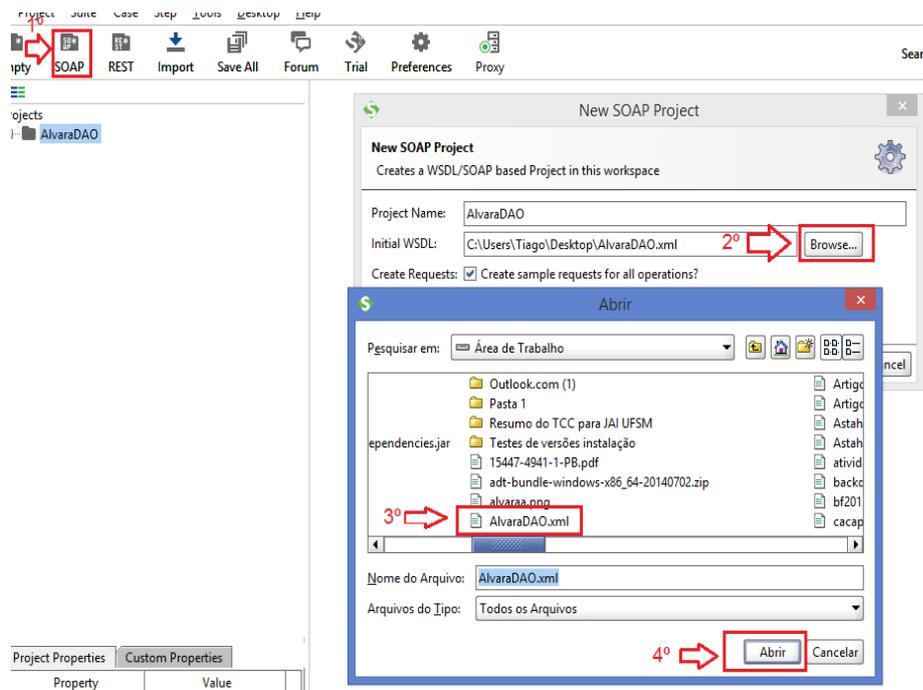


Figura 53 – Inserção de dados.

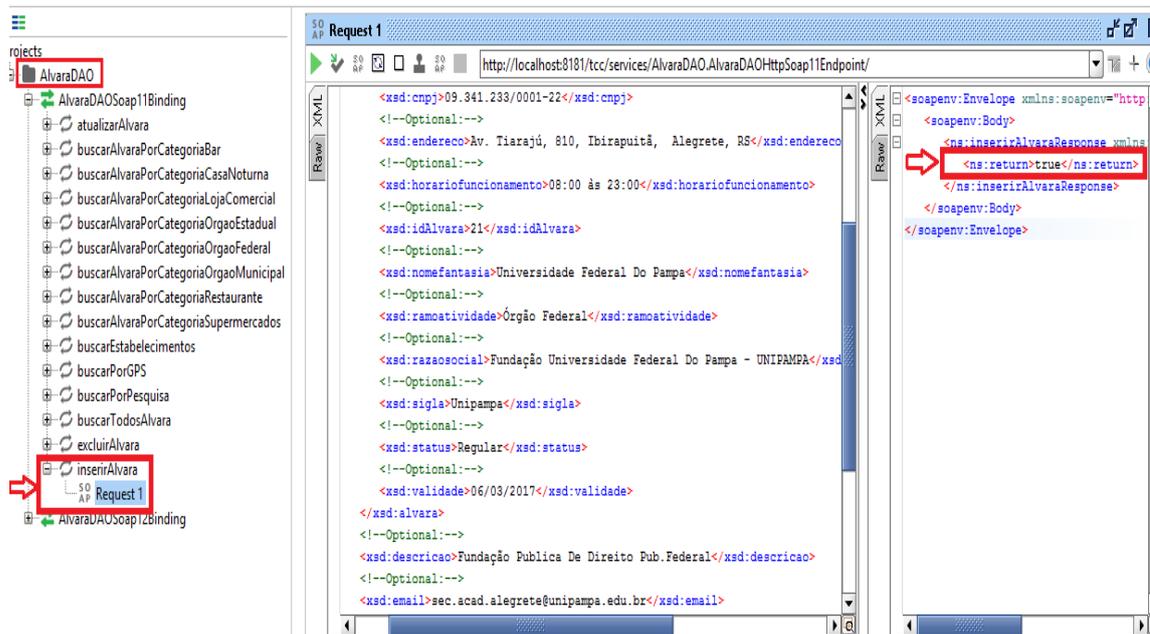


Figura 54 – Atualização de dados.

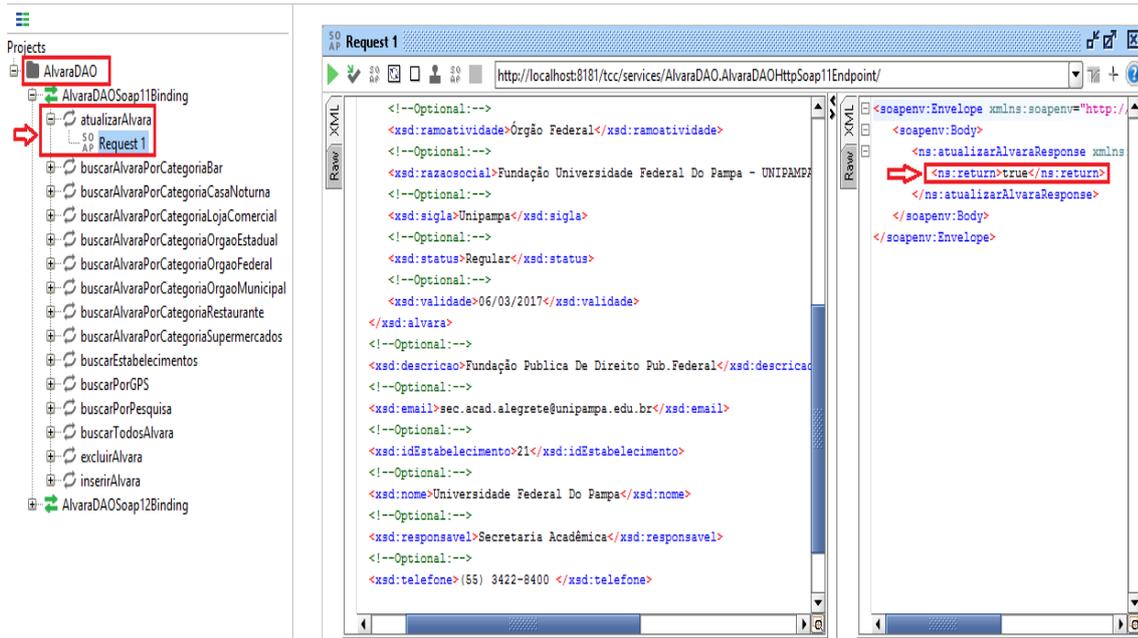


Figura 55 – Exclusão de dados.

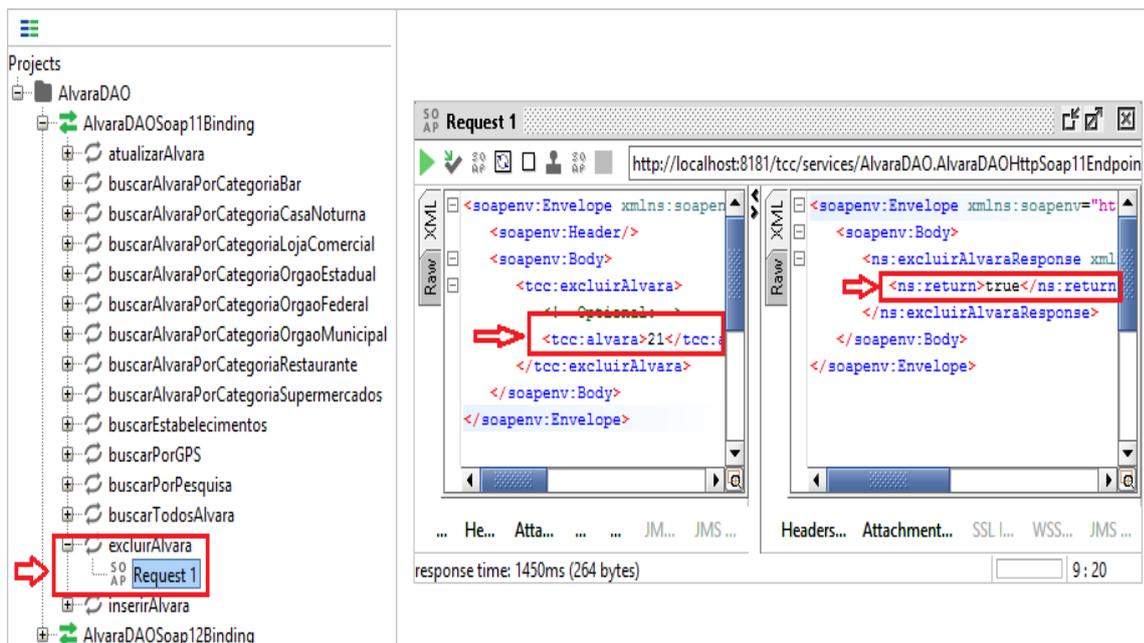


Figura 56 – Pesquisa Alvará por CNPJ.

The screenshot displays the SoapUI interface. On the left, the 'Projects' tree shows the 'AlvaraDAO' project with various service methods. The 'buscarPorPesquisa' method is selected, and a 'Request 1' is visible. The main window shows the raw XML of the request and response.

Request XML:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <tcc:buscarPorPesquisa>
      <!--Optional:-->
      <tcc:nomefantasia?/><tcc:nomefantasia>
      <!--Optional:-->
      <tcc:sigla?/><tcc:sigla>
      <!--Optional:-->
      <tcc:cnpj>09.341.233/0001-22/<tcc:cnpj>
    </tcc:buscarPorPesquisa>
  </soapenv:Body>
</soapenv:Envelope>
```

Response XML:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:buscarPorPesquisaResponse xmlns:ns="http://tcc">
      <ns:return xsi:type="ax21:Alvara" xmlns:ax21="http://tcc/xs:ax21">
        <ax21:cnpj>09.341.233/0001-22/</ax21:cnpj>
        <ax21:endereco>Av. Tiazajú, 810, Ibirapuitã, Alegre
        <ax21:horariofuncionamento>08:00 às 23:00/</ax21:horariofuncionamento>
        <ax21:idAlvara>21/</ax21:idAlvara>
        <ax21:lotacao maxima>1300/</ax21:lotacao maxima>
        <ax21:nomefantasia>Universidade Federal Do Pampa/</ax21:nomefantasia>
        <ax21:ramoatividade>Órgão Federal/</ax21:ramoatividade>
        <ax21:razaosocial>Fundação Universidade Federal Do Pampa/</ax21:razaosocial>
        <ax21:sigla>Unipampa/</ax21:sigla>
        <ax21:status>Regular/</ax21:status>
        <ax21:validade>06/03/2017/</ax21:validade>
      </ns:return>
    </ns:buscarPorPesquisaResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Figura 57 – Pesquisa Alvará por Sigla.

The screenshot displays the SoapUI interface. On the left, the 'Projects' tree shows the 'AlvaraDAO' project with various service methods. The 'buscarPorPesquisa' method is selected, and a 'Request 1' is visible. The main window shows the raw XML of the request and response.

Request XML:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <tcc:buscarPorPesquisa>
      <!--Optional:-->
      <tcc:nomefantasia?/><tcc:nomefantasia>
      <!--Optional:-->
      <tcc:sigla>Unipampa/<tcc:sigla>
      <!--Optional:-->
      <tcc:cnpj?/><tcc:cnpj>
    </tcc:buscarPorPesquisa>
  </soapenv:Body>
</soapenv:Envelope>
```

Response XML:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:buscarPorPesquisaResponse xmlns:ns="http://tcc">
      <ns:return xsi:type="ax21:Alvara" xmlns:ax21="http://tcc/xs:ax21">
        <ax21:cnpj>09.341.233/0001-22/</ax21:cnpj>
        <ax21:endereco>Av. Tiazajú, 810, Ibirapuitã, Alegre,
        <ax21:horariofuncionamento>08:00 às 23:00/</ax21:horariofuncionamento>
        <ax21:idAlvara>21/</ax21:idAlvara>
        <ax21:lotacao maxima>1300/</ax21:lotacao maxima>
        <ax21:nomefantasia>Universidade Federal Do Pampa/</ax21:nomefantasia>
        <ax21:ramoatividade>Órgão Federal/</ax21:ramoatividade>
        <ax21:razaosocial>Fundação Universidade Federal Do Pampa/</ax21:razaosocial>
        <ax21:sigla>Unipampa/</ax21:sigla>
        <ax21:status>Regular/</ax21:status>
        <ax21:validade>06/03/2017/</ax21:validade>
      </ns:return>
    </ns:buscarPorPesquisaResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Figura 58 – Pesquisa Alvará por Nome Fantasia.

The screenshot displays the SoapUI interface. On the left, the 'Projects' tree shows the 'AlvaraDAO' project with various service methods. The 'buscarPorPesquisa' method is selected, and a 'Request 1' is visible. The main window shows the raw XML of the request and response.

Request XML:

```

<env:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ttc:buscarPorPesquisa>
      <!--Optional:-->
      <ttc:nomefantasia>Universidade Federal Do Pampa</ttc:nomefantasia>
      <!--Optional:-->
      <ttc:sigla?/></ttc:sigla>
      <!--Optional:-->
      <ttc:cnpj?/></ttc:cnpj>
    </ttc:buscarPorPesquisa>
  </soapenv:Body>
</env:Envelope>

```

Response XML:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:buscarPorPesquisaResponse xmlns:ns="http://ttc">
      <ns:return xsi:type="ax21:Alvara" xmlns:ax21="http://www.w3.org/2001/XMLSchema-instance">
        <ax21:cnpj>09.341.233/0001-22</ax21:cnpj>
        <ax21:endereco>Av. Tiarajú, 810, Ibirapuitã, A
        <ax21:horariofuncionamento>08:00 às 23:00</ax21:horariofuncionamento>
        <ax21:idAlvara>21</ax21:idAlvara>
        <ax21:lotacaomaxima>1300</ax21:lotacaomaxima>
        <ax21:nomefantasia>Universidade Federal Do Pampa</ax21:nomefantasia>
        <ax21:ramoatividade>Órgão Federal</ax21:ramoatividade>
        <ax21:razaosocial>Fundação Universidade Federal
        <ax21:sigla>Unipampa</ax21:sigla>
        <ax21:status>Regular</ax21:status>
        <ax21:validade>06/03/2017</ax21:validade>
      </ns:return>
    </ns:buscarPorPesquisaResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

Figura 59 – Pesquisa do alvará por categoria.

The screenshot displays the SoapUI interface. On the left, the 'Projects' tree shows the 'AlvaraDAO' project with various service methods. The 'buscarAlvaraPorCategoriaOrgaoFederal' method is selected, and a 'Request 1' is visible. The main window shows the raw XML of the request and response.

Request XML:

```

<env:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ttc:buscarAlvaraPorCategoriaOrgaoFederal/>
  </soapenv:Body>
</env:Envelope>

```

Response XML:

```

<ns:return xsi:type="ax21:Alvara" xmlns:ax21="http://www.w3.org/2001/XMLSchema-instance">
  <ax21:validade>28/02/2017</ax21:validade>
</ns:return>
<ns:return xsi:type="ax21:Alvara" xmlns:ax21="http://www.w3.org/2001/XMLSchema-instance">
  <ax21:cnpj>09.341.233/0001-22</ax21:cnpj>
  <ax21:endereco>Av. Tiarajú, 810, Ibirapuitã, Alegrete, R
  <ax21:horariofuncionamento>08:00 às 23:00</ax21:horariofuncionamento>
  <ax21:idAlvara>21</ax21:idAlvara>
  <ax21:lotacaomaxima>1300</ax21:lotacaomaxima>
  <ax21:nomefantasia>Universidade Federal Do Pampa</ax21:nomefantasia>
  <ax21:ramoatividade>Órgão Federal</ax21:ramoatividade>
  <ax21:razaosocial>Fundação Universidade Federal Do Pampa
  <ax21:sigla>Unipampa</ax21:sigla>
  <ax21:status>Regular</ax21:status>
  <ax21:validade>06/03/2017</ax21:validade>
</ns:return>
<ns:return xsi:type="ax21:Alvara" xmlns:ax21="http://www.w3.org/2001/XMLSchema-instance">

```

Figura 60 – Pesquisa do alvará por GPS.

The screenshot shows the SoapUI interface with the following details:

- Projects:** AlvaraDAO > AlvaraDAOSoap11Binding > buscarPorGPS (highlighted with a red box and arrow).
- Request 1:**

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <tcc:buscarPorGPS>
      <!--Optional:-->
      <tcc:sigla>Unipampa</tcc:sigla>
    </tcc:buscarPorGPS>
  </soapenv:Body>
</soapenv:Envelope>

```
- Response:**

```

<ns:buscarPorGPSresponse xmlns:ns="http://tcc/">
  <ns:return xsi:type="ax21:Alvara" xmlns:ax21="http://tcc/xsd" xmlns:ax211="http://www.w3.org/2001/XMLSchema-instance">
    <ax21:cmpj>09.341.233/0001-22</ax21:cmpj>
    <ax21:endereco>Av. Tiarajú, 810, Ibirapuitã, Alegrete, RS</ax21:endereco>
    <ax21:horariofuncionamento>08:00 às 23:00</ax21:horariofuncionamento>
    <ax21:idAlvara>21</ax21:idAlvara>
    <ax21:lotacao maxima>1300</ax21:lotacao maxima>
    <ax21:nome fantasia>Universidade Federal Do Pampa</ax21:nome fantasia>
    <ax21:ramoatividade>Órgão Federal</ax21:ramoatividade>
    <ax21:razaosocial>Fundação Universidade Federal Do Pampa - UNIPAMPA</ax21:razaosocial>
    <ax21:sigla>Unipampa</ax21:sigla>
    <ax21:status>Regular</ax21:status>
    <ax21:validade>06/03/2017</ax21:validade>
  </ns:return>

```

Figura 61 – Pesquisa dos dados dos contatos alvará.

The screenshot shows the SoapUI interface with the following details:

- Projects:** AlvaraDAO > AlvaraDAOSoap11Binding > buscarEstabelecimentos (highlighted with a red box and arrow).
- Request 1:**

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <tcc:buscarEstabelecimento>
  </tcc:buscarEstabelecimento>
  </soapenv:Body>
</soapenv:Envelope>

```
- Response:**

```

<ns:return xsi:type="ax21:Estabelecimento" xmlns:ns="http://tcc/" xmlns:ax21="http://www.w3.org/2001/XMLSchema-instance">
  <ax21:alvara xsi:nil="true"/>
  <ax21:descricao>Fundação Publica De Direito Pub.Federal</ax21:descricao>
  <ax21:email>sec.acad.alegrete@unipampa.edu.br</ax21:email>
  <ax21:idEstabelecimento>21</ax21:idEstabelecimento>
  <ax21:nome>Universidade Federal Do Pampa</ax21:nome>
  <ax21:responsavel>Secretaria Acadêmica</ax21:responsavel>
  <ax21:telefone>(56) 3422-8400</ax21:telefone>
</ns:return>

```

At the bottom of the window, it shows: response time: 279ms (10757 bytes) and 195:10.

APÊNDICE J – Apêndice J

Figura 62 – Testes na versão 2.3.7 GingerBread

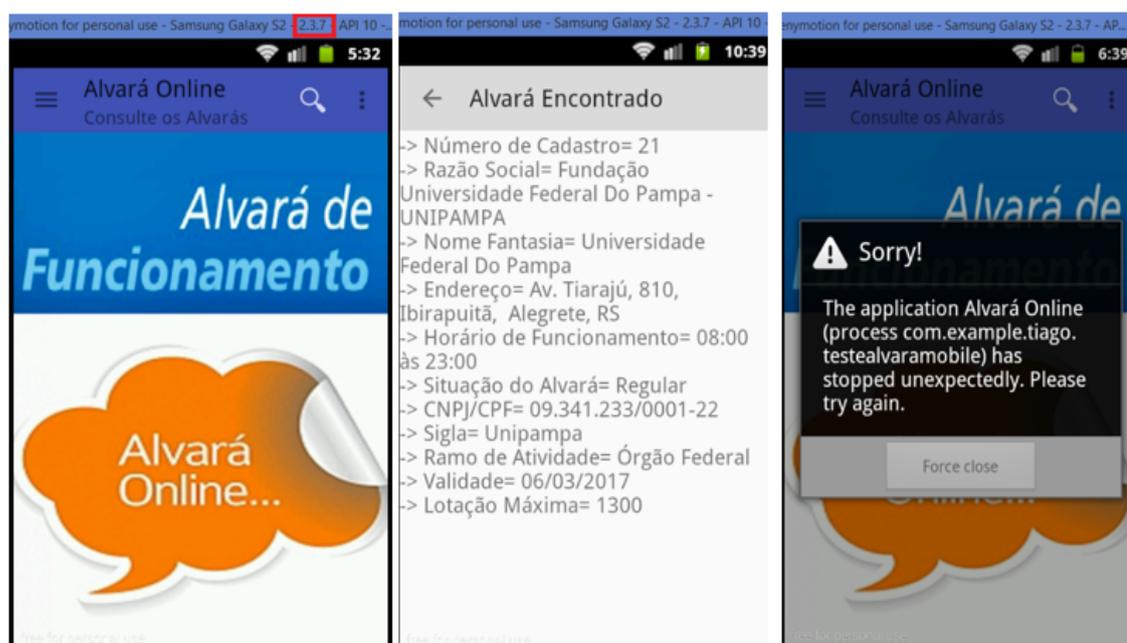


Figura 63 – Testes na versão Android 4.1.1 JellyBean

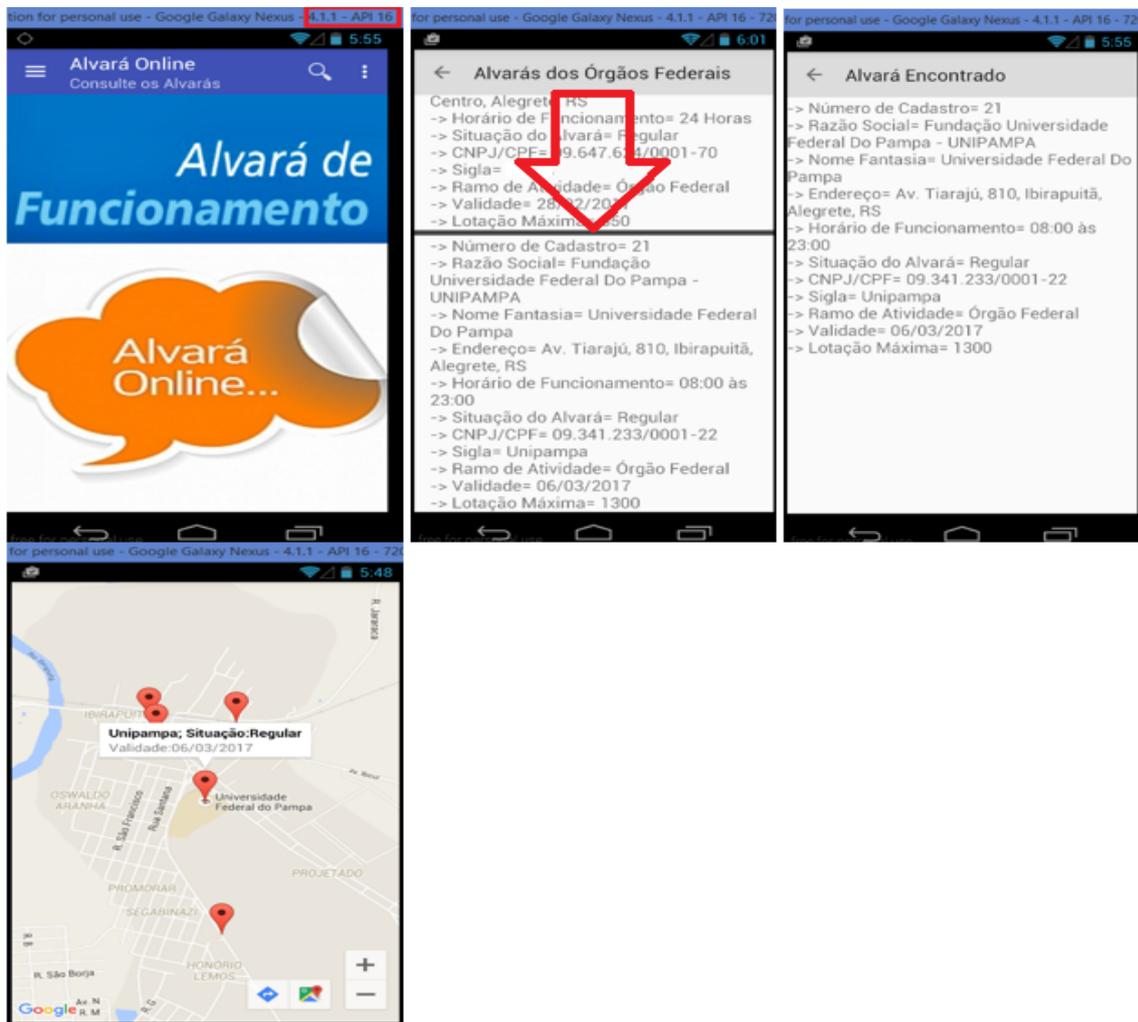


Figura 64 – Testes na versão Android 4.2.2 JellyBean

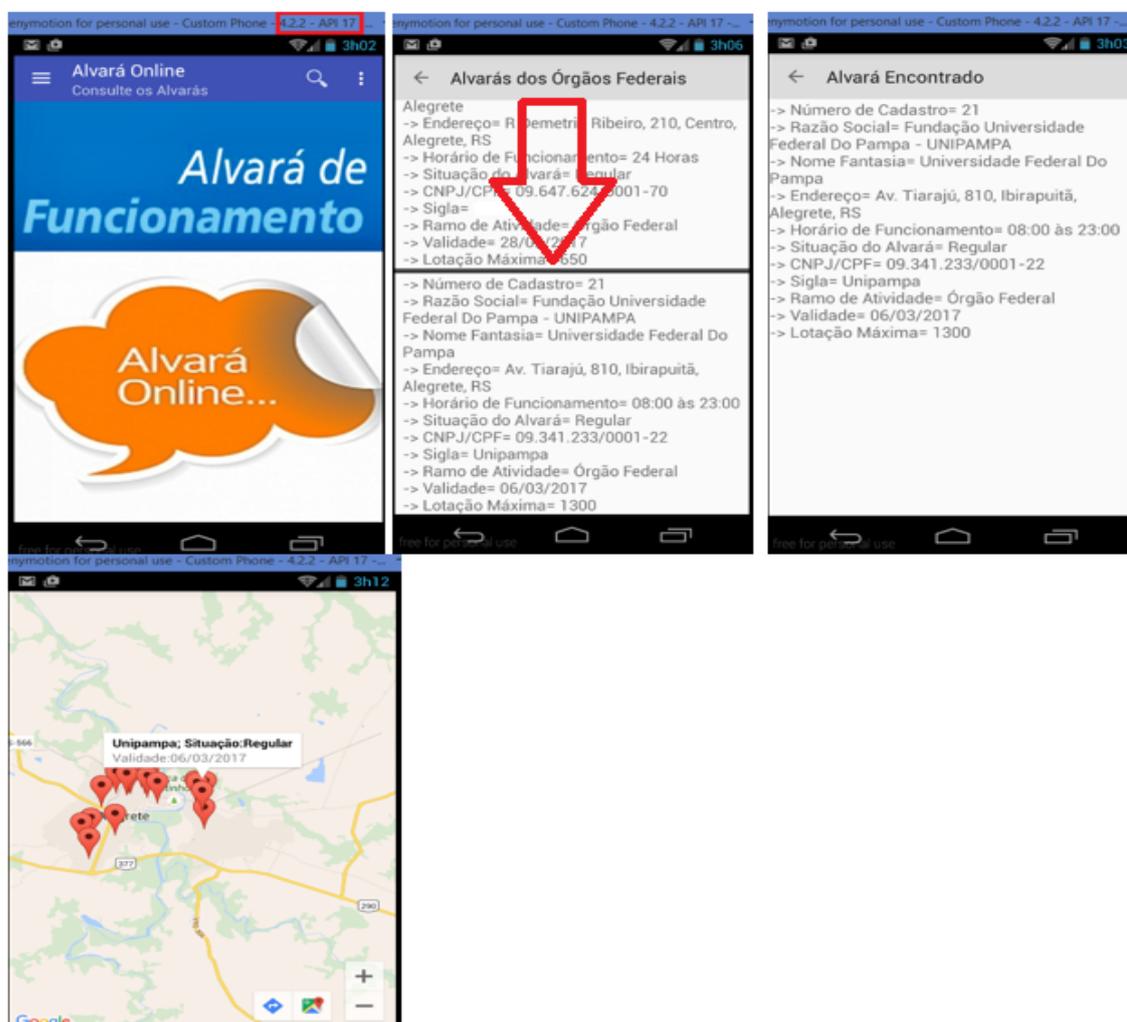


Figura 65 – Testes na versão Android 4.3 JellyBean

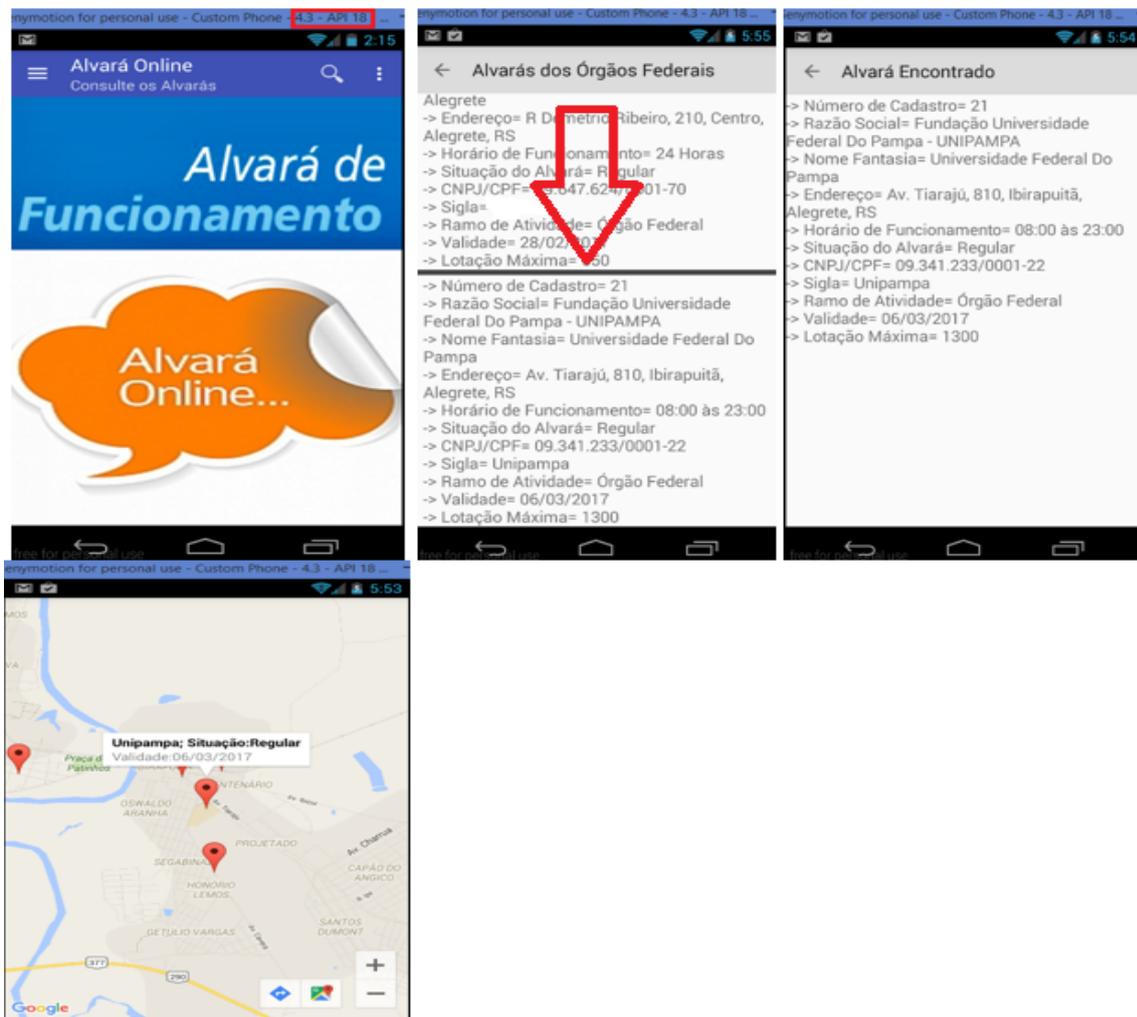


Figura 66 – Testes na versão Android 4.4.4 KitKat

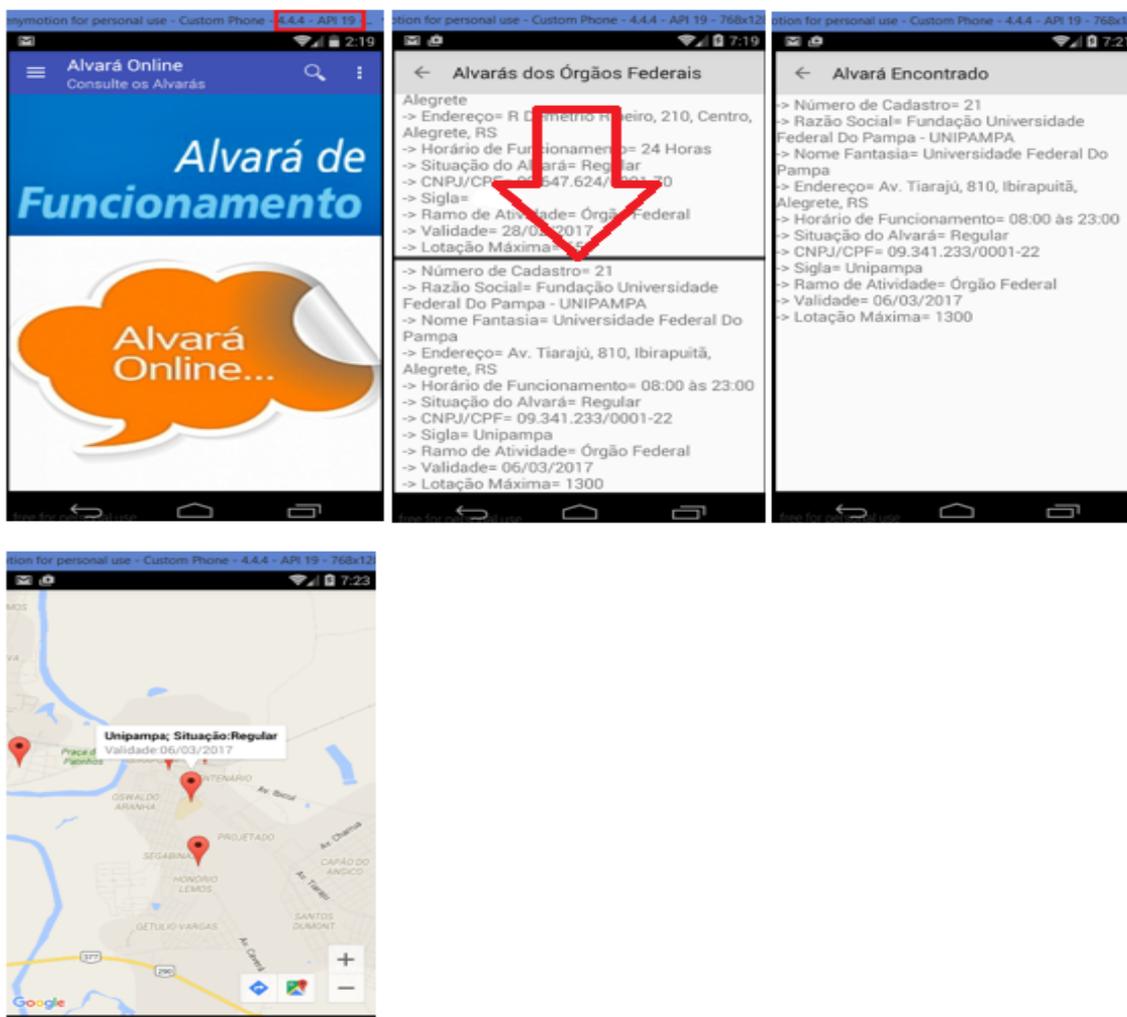


Figura 67 – Testes na versão Android 5.0.0 Lollipop

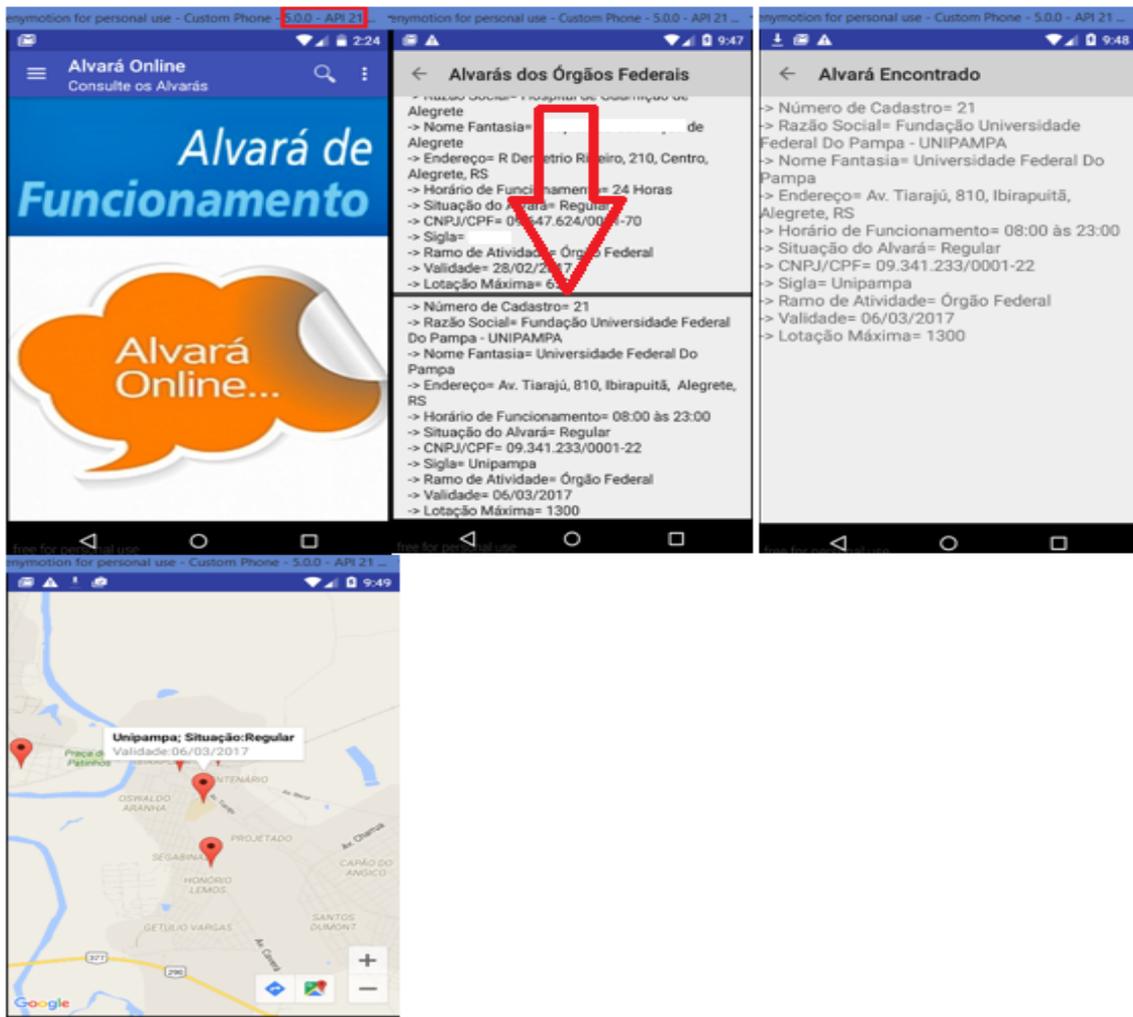


Figura 68 – Testes na versão Android 5.1.0 Lollipop

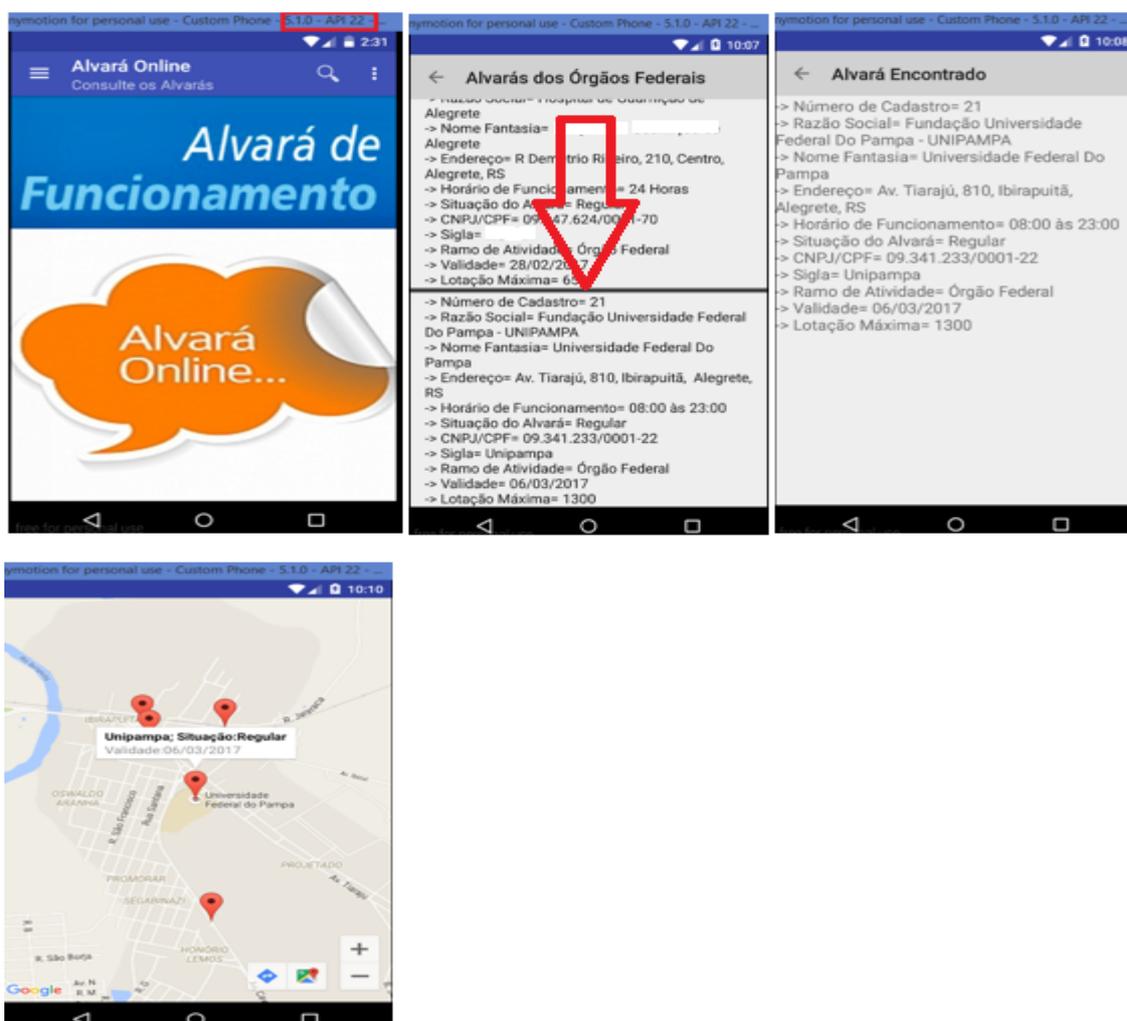


Figura 69 – Testes na versão Android 6.0.0 Marshmallow

